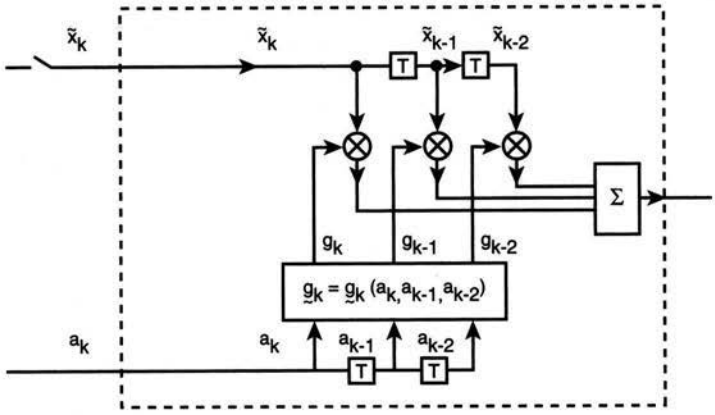
Figure 12.11 The General Structure of the Baud Rate Timing Recovery Circuit



Specifically, we can have a simple

$$F_1(\bar{x}_k(\tau), \bar{a}_k(\tau)) = \frac{1}{2}(x_k(\tau)a_{k-1}(\tau) - x_{k-1}(\tau)a_k(\tau)) / E\{a_k^2(\tau)\}$$

as illustrated in Figure 12.12 for $f_1(\tau)$ in which $E\{a_k^2(\tau)\}$ is available through the automatic gain control circuits, and a simple

$$F_2(\bar{x}_k(\tau), \bar{a}_k(\tau)) = a_{k-1}(\tau)(x_k(\tau) - a_k(\tau)h_0(\tau)) / E\{a_k^2(\tau)\}$$

as illustrated in Figure 12.13 for $f_2(\tau)$ in which $h_0(\tau)$ is also available through the received signal strength.

Figure 12.12 A Simple Timing Circuit for Even Channel Impulse Response

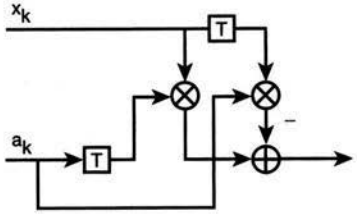
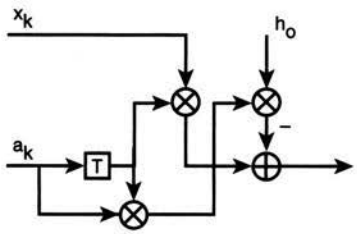


Figure 12.13 A Simple Timing Circuit for Odd Channel Impulse Response

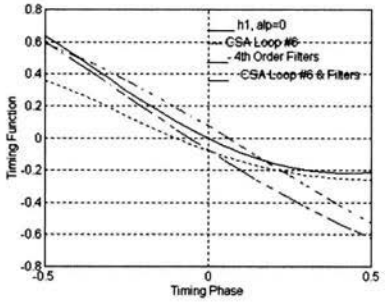


With proper recognition of

$$f_1(\tau) = \frac{1}{2}\big(h(\tau + T) - h(\tau - T)\big)$$

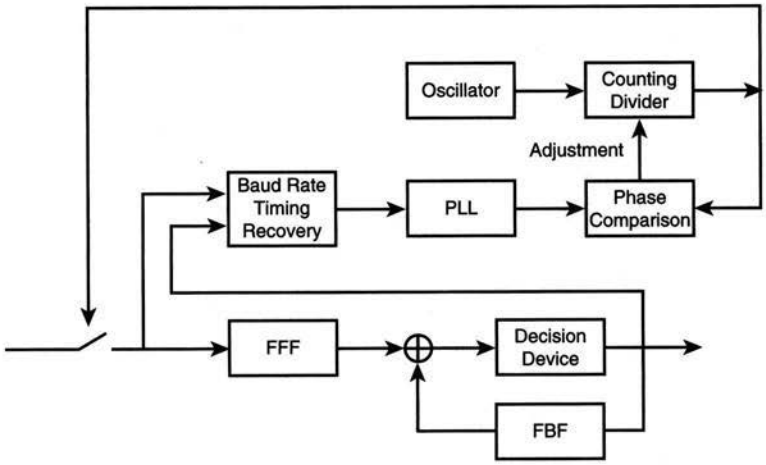
that is, the selection of received signal samples in relation to corresponding estimated data symbol, this timing recovery technique can also be applied to a general twisted-pair telephone subscriber loop channel, including transmitter and receiver filters. With proper delay, Figure 12.14 shows the timing function for a zero-length loop with a fourth-order Butterworth transmit filter, a fourth-order receiver filter, and a 9 kft, 26-gauge loop with and without transmit and receiver filters in comparison with that of a Nyquist channel.

Figure 12.14 Timing Function on Twisted-Pair Loops



Due to the extensive channel intersymbol interference distortion, the estimation of the data symbols required in the timing function can only be obtained after the decision feedback channel equalization. Figure 12.15 shows the general structure of an HDSL timing recovery circuit.

Figure 12.15 An HDSL Timing Recovery Circuit



The operation of this timing recovery circuit is based on the assumption that the frequency difference between the transmit clock and the receiver clock is so close (< 30 PPM) that the convergence of the equalizer can be realized even without the timing recovery circuit.

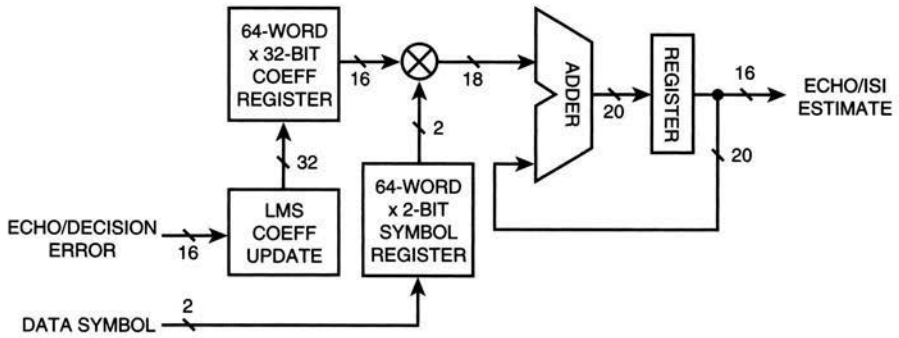
The main purposes of the timing recovery circuit are to obtain the optimal sampling phase and to avoid long-term frequency/phase shift between transmit and receiver clocks. There could be some jitter from the original baud rate timing recovery circuit. The inclusion of a Phase Locked Loop (PLL) should help improve the overall timing recovery performance. A PLL consists of a very narrow bandpass filter, which only allows the desired frequency to pass through.

12.2.5 HDSL DFE/EC Semiconductor Technologies

Two major signal processing functions of an HDSL transceiver are the adaptive Decision Feedback channel Equalizer (DFE) and the adaptive Echo Canceller (EC). The DFE consists of a *Feedforward Filter* (FFF) and a *Feedback Filter* (FBF). The FFF is an adaptive *Finite Impulse Response* (FIR) filter with eight taps and 12 bits, corresponding to the ADC precision, in the data path. The FBF and EC are both adaptive FIR filters with about 128 taps. The data path for both FFF and EC is two bits for the 2B1Q four level signal.

Either the FBF or EC function can be effectively implemented with an *Application Specific Integrated Circuit* (ASIC) of the same structure. A possible ASIC chip implementation with a 64-tap cascading taps is explained in the following paragraphs.[72] Figure 12.16 shows a simplified block diagram of the EC/DFE ASIC chip.

Figure 12.16 Simplified Block Diagram of EC/DFE Chip



The coefficients of the adaptive FIR filter are updated using a sign-LMS update equation given by

$$C_k(n+1) = C_k(n) + \mu e(n)\mathrm{sgn}\big(x(n-k)\big)$$

for $k = 1, 2, \ldots 64$. The sign algorithm and the true LMS update algorithm have virtually identical performance in a 2BIQ system. This is due to the fact that $x(n–k)$ is a two-bit signal $(+3, +1, -1, -3)$ and, therefore, quantizing $x(n–k)$ to 1 bit $(+1, -1)$ has virtually no effect on the convergence of the algorithm. This modification does, however, result in a very simple multiplierless hardware implementation when power-of-two step-sizes are used.

The FBF and EC update equations have the same form. In an EC application, $x(n)$ represents the transmit symbol sequence and $e(n)$ represents the residual uncancelled echo error. In an FBF application, $x(n)$ represents the receive symbol decisions and $e(n)$ represents the slicer decision error. Figure 12.17 shows an overview of the EC/DFE chip architecture.

The EC/DFE chip accepts a two-bit data symbol, $x(n)$, and a 16-bit error input, $e(n)$, and produces a 16-bit echo estimate (for EC applications) or equalized signal (for DFE applications), $y(n)$, during each symbol period. In addition, all filter coefficients are updated in every symbol period using a sign-LMS algorithm. The 16-bit error input is either the residual echo error (for EC applications) or the decision error (for DFE applications). The FIR filter coefficients are updated and stored with 32 bits of precision, and the filter computations are performed to 20 bits of accuracy.

For applications requiring filters longer than 64 taps, the EC/DFE chip can be cascaded to implement adaptive FIR filters of arbitrary length. Figure 12.18 illustrates how to interconnect several EC/DFE chips.

Figure 12.17 Overview of EC/DFE Chip Architecture

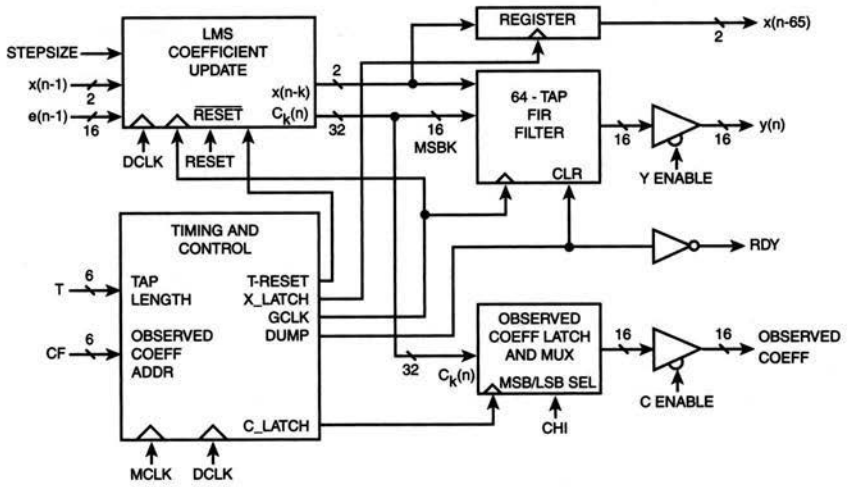
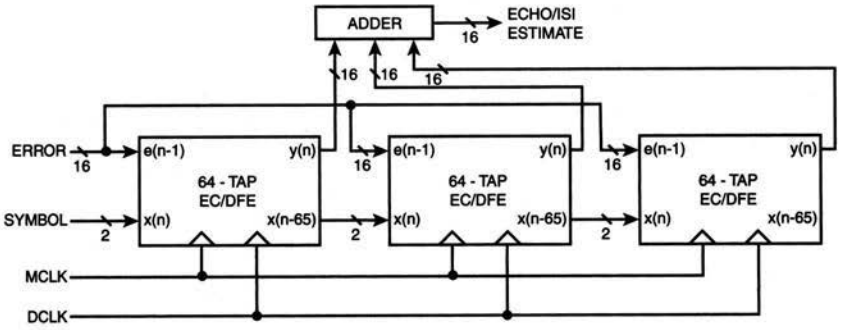


Figure 12.18 Interconnections Required for Cascading EC/DFE Chips

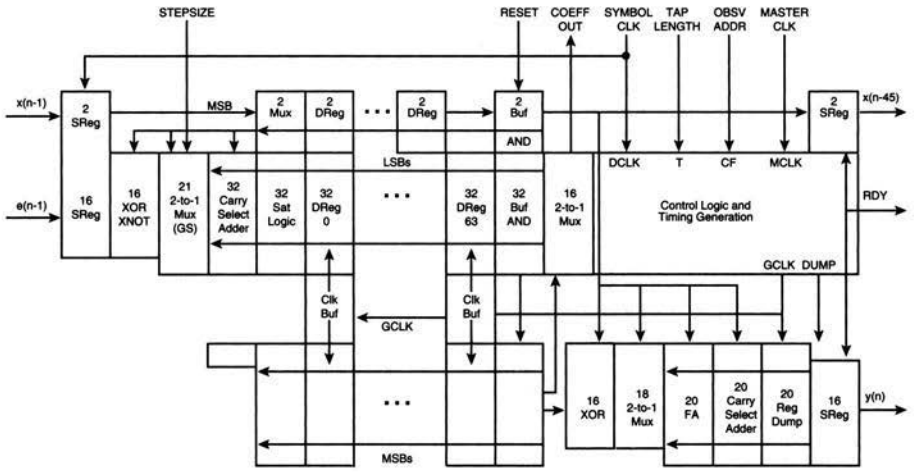


The EC/DFE chip consists of four main sections:

- An FIR filter section
- An LMS coefficient update section
- A register block
- A control section

The chip floor plan exploits the regular nature of these elements by using a parallel datapath approach. Figure 12.19 shows the floorplan of the EC/DFE chip.

Figure 12.19 Floor Plan of EC/DFE Chip



The data flow is primarily in one direction, as follows:

1. The error input is accepted on the left side of the EC/DFE chip and is routed directly to the LMS coefficient update section.
2. The updated coefficients are then directly routed to the input of the register block.
3. The input symbol data is also routed directly to the register block.
4. The filter coefficients and symbol data at the output of the register block are then delivered to the FIR filter section.
5. The final filter output is delivered to the output pads on the right side of the chip.

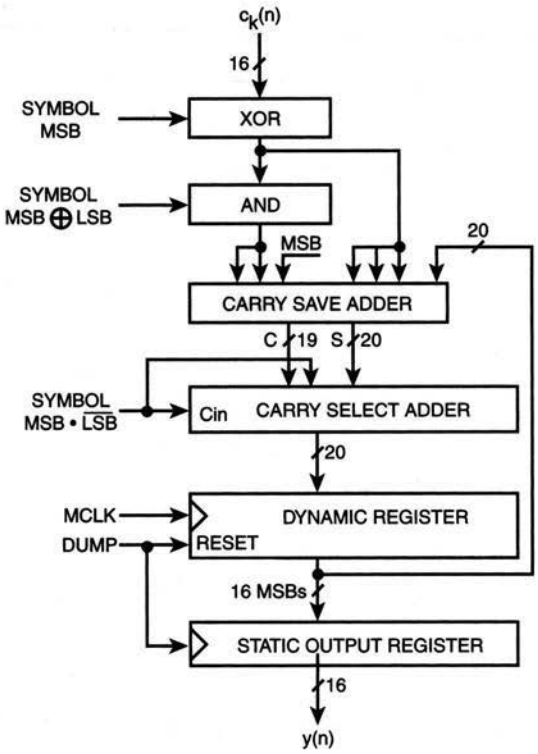
The FIR Filter Section

The FIR filter section is essentially a multiply-accumulate block that operates at the master clock rate and implements the following equation:

$$y(n) = \sum C_k x(n-k)$$

Figure 12.20 shows a block diagram of the FIR filter section.

Figure 12.20 FIR Filter Section Block Diagram



The two major logic blocks in the FIR filter are the *multiplier* and the *accumulator*. However, since the symbol word length is only two bits, the multiplier simplifies to a shift-and-add block. To achieve the speed goals, the multiplier was implemented using a carry-save adder and the accumulator was implemented using a carry-select adder.[73] This arrangement minimizes the delay through the FIR block since only one 20-bit carry-propagate addition is needed per multiply-accumulate cycle.

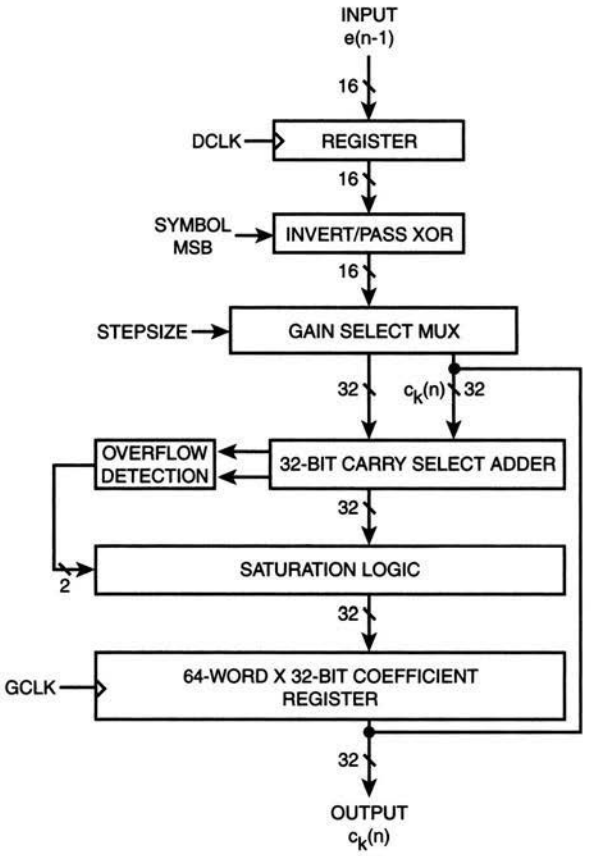
The LMS Coefficient Update Section

The LMS coefficient update section implements the sign-LMS algorithm as specified by

$$C_k(n+1) = C_k(n) + \mu e(n-1)\mathrm{sgn}\left[x(n-k-1)\right]$$

The coefficient update equation can be mapped into a straightforward bit-slice architecture, as shown in Figure 12.21.

Figure 12.21 LMS Update Section Block Diagram



The input error signal *e(n – 1)* is clocked into a static register at the symbol rate. The multiplication by the sign of the data symbol is achieved with an invert-or-pass circuit (an array of XORs), and the multiplication by the power-of-two step-size μ is achieved by a hardwired shift. The value of μ is user selectable to either of two values through the use of a two-to-one multiplexer.

Finally, a 32-bit carry-select adder is used to sum the update term $\mu e(n-1)\text{sgn}[x(n-k-1)]$ with the old coefficient value, with the result being stored in the register block. After all 64 coefficients have been updated, a new error input is received and the cycle is repeated.

The Register Block

The register block consists of a 64-word by 34-bit wide shift register stack. Thirty-two bits of each word are allocated for coefficient storage, and two bits are allocated for symbol data

storage. A dynamic master-slave, eight-transistor clocked-CMOS (C^2MOS) shift register cell design[74] was used to achieve high-speed operation. The design has the drawback, however, of drawing much more power than a conventional RAM-based implementation because all data is shifted on every clock cycle.

If a RAM design were used, only one column of 34-bit data would be accessed and written to in each clock cycle. In spite of this drawback, the shift register design was implemented in the EC/DFE chip because speed and robustness were more important design criteria than power dissipation for this demonstration chip.

Note
Dynamic RAM cell designs have the potential for significantly reducing the area and power consumption of the register block, although meeting the speed goals will be more difficult.

The filter coefficients that are stored in the register block are delivered to the LMS coefficient update section, where they are updated and circulated back into the register block. Because undefined states are initially present in the register stack (after power-up), a means of resetting the coefficients and symbol data is provided. Regardless of the desired tap length of the FIR filter, all coefficients and symbol data are cycled through the register block.

The Control Section

The control section generates the internal system clock and various control signals for the FIR filter section, the LMS coefficient update section, and the register block. The control section also generates the necessary control signals to accommodate variable tap lengths and to allow complete coefficient observability.

For tap length programmability, the desired tap length of the FIR filter is input to the EC/DFE chip as a six-bit address. For coefficient observability, a 6-bit address is input to the EC/DFE chip to select the appropriate 32-bit coefficient, and a 1-bit select signal determines whether the 16 *Most Significant Bits* (MSB) or 16 *Least Significant Bits* (LSB) of the coefficient are delivered to the 16-bit output bus.

12.3 Performance Objectives

The performance of HDSL is primarily determined by the selected loop population and corresponding transceiver architecture. The value of termination impedance also has a slight

influence on the performance of HDSL. The issue of the slight transmission throughput difference in opposite directions for HDSL also needs a practical solution.

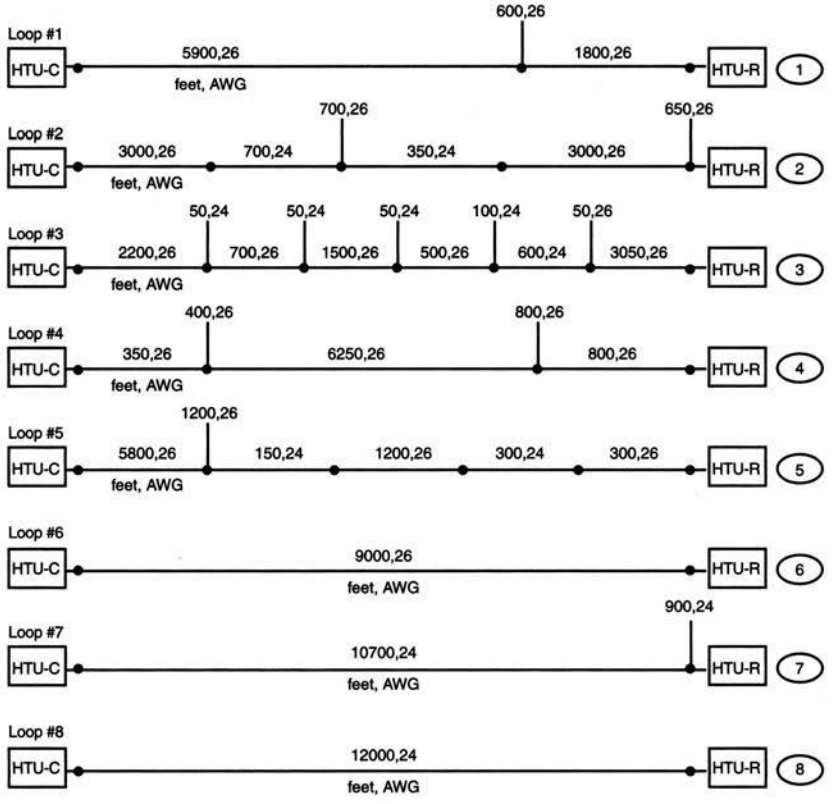
12.3.1 General HDSL Performance Objectives

The performance of the HDSL system is primarily determined by the following factors:

- Transmission characteristics of the twisted-pair telephone loop
- The receiver front-end noise level
- The structure of the HDSL transceiver

Figure 12.22 shows configurations of eight selected *Carrier Serving Area* (CSA) test loops[75] that have been used in the engineering process of the HDSL system.

Figure 12.22 CSA Test Loops



The transmission characteristics of these CSA loops have been used in a computer simulation study, incorporated into the test procedure,[76] and duplicated by test equipment.[77,78] The worst-case CSA loops are typically defined by a straight 9 kft, 26-gauge loop or a straight 12 kft, 24-gauge loop. For loops with bridge taps, the total length including bridge taps should be less than 9 kft or 12 kft, depending on the cable gauge.

Figures 12.23a and 12.23b show the transfer functions of the eight CSA test loops. These transfer functions are characterized by a 10 dB loss at about 1 kHz and about 35 dB loss at about 200 kHz. Bridged taps cause nulls at different frequencies.

Figure 12.23a Transfer Function of CSA Loops 1–4

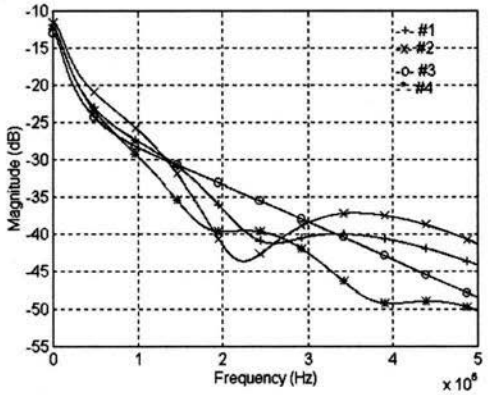
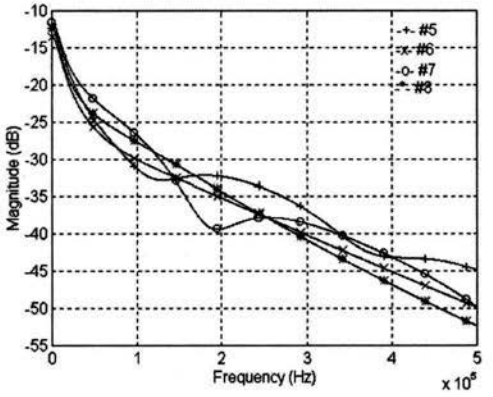
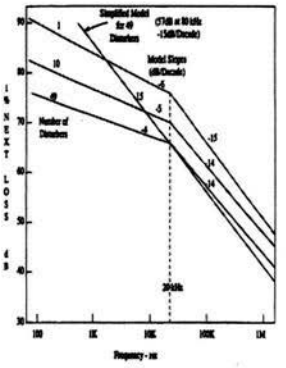


Figure 12.23b Transfer Function of CSA Loops 5–8



Because of the use of an echo-cancellation-based full-duplex method, the dominant noise component at the receiver front end is the self Near End Crosstalk, self-NEXT. Self-NEXT is induced by neighboring HDSL transmitters. The self-NEXT noise level depends on the transmit signal level and the NEXT loss. Figure 12.24 shows the NEXT loss model for three different numbers of disturbers and one simplified NEXT model for 49 disturbers. For HDSL simulation studies and the performance test environment, the simplified 49-disturber NEXT model is used. The 49 disturber is the worst-case scenario, while the use of a straight line extension at low frequencies introduces little performance difference.

Figure 12.24 NEXT Model



The performance of an HDSL transceiver can be judged by the SNR of its decision feedback channel equalizer output. The SNR in turn can be expressed by the mean squared error of the decision feedback channel equalizer as

$$SNR = \frac{1 - E\left[e_k^2\right]}{E\left[e_k^2\right]} \approx \frac{1}{E\left[e_k^2\right]}$$

The mean squared error of the HDSL decision feedback channel equalizer is

$$E\left[e_k^2\right] = E\left[\left(d_k - X_k^T A_{opt} + D_{k-1}^T B_{opt}\right)^2\right]$$

The values are as follows:

d_k is the desired data symbol.

$X_k^T = [x_k \quad x_{k-1} \quad \ldots \quad x_{x-n+1}]$ is the received signal vector.

$A_{opt}^T = [a_1 \quad a_2 \quad \ldots \quad a_n]_{opt}$ is the feedforward filter coefficient vector.

$D_{k-1}^T = [d_{k-1} \quad d_{k-2} \quad \ldots \quad d_{x-m}]$ is the estimated data symbol vector.

$B_{opt}^T = [b_1 \quad b_2 \quad \ldots \quad b_m]_{opt}$ is the feedback filter coefficient vector.

e_k is the error between the desired signal and the decision feedback equalizer output.

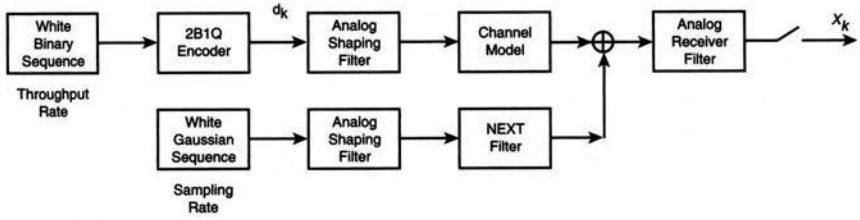
A_{opt}^T and B_{opt}^T are defined as

$$A_{opt}^T = \left(E[X_k X_k^T] - E[X_k D_{k-1}^T]E[D_{k-1} X_k^T]\right)^{-1} E[d_k X_k^T]$$

$$B_{opt}^T = A_{opt}^T E[X_k D_{k-1}^T]$$

The performance of an HDSL transceiver can be estimated through computer simulation based on a pseudo-random signal generated according to the configuration, as illustrated in Figure 12.25.

Figure 12.25 The Generation of an HDSL Received Signal



A white binary sequence is generated and encoded as a random data symbol sequence. The binary sequence is generated at the data transmission rate and encoded into the symbol rate. Meanwhile, a white Gaussian sequence is also generated to drive the NEXT filter. The white Gaussian sequence is generated at the simulation sampling rate. The simulation sampling rate is usually 5 to 10 times that of the baud rate to emulate the channel covering excessive bandwidth signal energy and to show the effect of sampling phase.

The performance of an HDSL transceiver is characterized by the relationship between the received signal vector, X_k, and the transmit data symbol vector, D_k. The data symbol vector, D_k, is directly available after the 2B1Q encoder. The received signal vector is the combination of the data symbol after the filtering of the equivalent channel consisting of the transmit filter, the channel, and the receiver filter, and the white Gaussian sequence after the filtering of the equivalent NEXT coupling function consisting of the transmit filter, the NEXT filter, and the receiver filter.

Both transmit and receiver filters are fourth-order Butterworth filters with a 3 dB corner frequency of 200 kHz. The channel models are these eight selected CSA loops. The NEXT filter is an FIR filter designed according to the simplified 49-disturber NEXT model. Figure 12.26 shows the impulse response of the NEXT FIR filter.

Figure 12.26 Impulse Response of the NEXT Filter

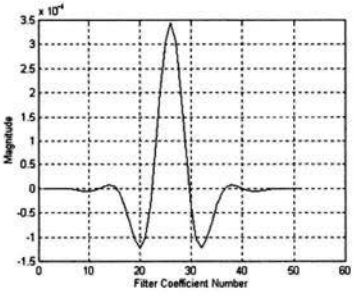


Figure 12.27 shows the transfer function of the NEXT FIR filter.

Figure 12.27 Transfer Function of the NEXT Filter

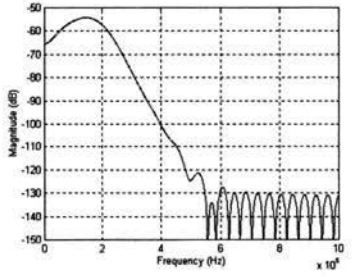


Table 12.4 shows simulated DEF output SNR for these eight test loops. The required SNR is 21.5 dB for an error rate of 10^{-7}. A 6 dB measured performance margin is required for all

HDSL test loops. The simulation results in Table 12.4 are based on a baud rate DFE with 16 FFF and 128 FBF coefficients. Results might be different for different DFE sizes and other hardware parameter assumptions.

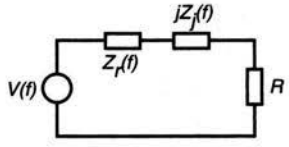
Table 12.4 Simulated Performance on Eight HDSL Test Loops

Test Loop Number	SNR	Margin
1	30.6	9.1
2	30.1	8.6
3	27.8	6.3
4	29.4	7.9
5	30.9	9.4
6	29.2	7.7
7	31.1	9.6
8	30.1	8.6

12.3.2 *The Effect of Termination Resistance Value*

The subscriber loop terminal resistor value has some effect on the performance of an HDSL transceiver. Traditionally, the terminal resistor value can be chosen to maximize the front-end received signal power as in some analog receivers. For the case of a receiver with an ideal DFE, variation in performance due to the terminal resistor could disappear because of the matching filter effect. In practice, the HDSL terminal resistor value can be adjusted to further improve the overall system performance when a baud rate DFE is utilized, as shown in Figure 12.28.

Figure 12.28 Terminal Resistance Optimization



For a frequency variable voltage source with a frequency-dependent source impedance, $Z(f) = Z_r(f) + jZ_j(f)$, we would like to find the optimal load resistor value to maximize the total power delivered from the source to the load.[79] We have the total power:

$$P = \int_0^\infty P_r(f)df = \int_0^\infty |I_r(f)|^2 Rdf = \int \frac{V^2(f)R}{\left(Z_r(f)+R\right)^2 + Z_j^2(f)} df$$

The values are as follows:

$V(f)$ is the frequency variable voltage.

$Z_r(f)$ is the real part of the frequency-dependent source impedance.

$Z_j(f)$ is the imaginary part of the frequency-dependent source impedance.

R is the load resistor.

Setting

$$\frac{dP}{dR} = 0$$

we get

$$\int \frac{\left(\left(Z_r(f)+R\right)^2 + Z_j^2(f)\right)V^2(f) - 2R\left(Z_r(f)+R\right)V^2(f)}{\left(\left(Z_r(f)+R\right)^2 + Z_j^2(f)\right)^2} df = 0$$

We can write the preceding expression as

$$\int_0^\infty \frac{\left(Z_r^2(f)+Z_j^2(f)\right)V^2(f)}{\left(\left(Z_r(f)+R\right)^2 + Z_j^2(f)\right)^2} df = R^2 \int_0^\infty \frac{V^2(f)}{\left(\left(Z_r(f)+R\right)^2 + Z_j^2(f)\right)^2} df$$

We can solve for R iteratively using the following:

$$R_{i+1}{}^2 = \frac{\displaystyle\int_0^\infty \frac{\left(Z_r^2(f)+Z_j^2(f)\right)V^2(f)}{\left(\left(Z_r(f)+R_i\right)^2 + Z_j^2(f)\right)^2} df}{\displaystyle\int_0^\infty \frac{V^2(f)}{\left(\left(Z_r(f)+R_i\right)^2 + Z_j^2(f)\right)^2} df}$$

For the HDSL system, we let $V^2(f) = P_5(f)$ in which

$$P_5(f) = K\frac{1}{f_0}\frac{\sin^2\left(\frac{\pi f}{f_0}\right)}{\left(\frac{\pi f}{f_0}\right)^2}\frac{1}{1+\left(\frac{f}{f_h}\right)^4}\frac{1}{1+\left(\frac{f_l}{f}\right)^4}$$

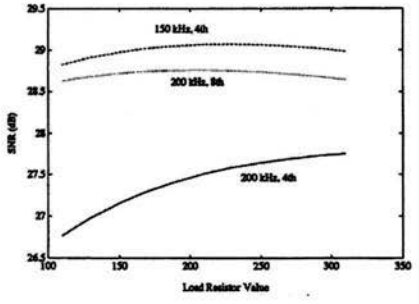
for

$f_0 = 400kHz$
$f_h = 256kHz$
$f_l = 5kHz$
$Z_r(f)$ and $Z_j(f)$ equal to the input impedance of a 9 kft, 26-gauge loop.

We find that $R_{opt} = 120\Omega$.

Computer simulations are carried out based on the 9 kft, 26-gauge loop and the 49-disturber self-NEXT model for the HDSL system. Figure 12.29 summarizes computer simulation results in terms of SNRs as functions of terminal resistor value R and lowpass filter frequency responses. The fourth and eighth labels in Figure 12.29 represent the orders of receiver filters.

Figure 12.29 DFE Output SNR Versus R for HDSL



The front-end received signal power can be maximized by adjusting the load resistor value; however, the SNR at the receiver detection point is not a direct function of the front-end

signal power. The SNR is affected by terminal resistor value, receiver lowpass filter frequency response, and the structure of the DFE. The effect of varying the terminal resistor value becomes small when the excessive crosstalk power due to the sampling image is suppressed by the receiver lowpass filter. A 135 ohms terminal resistor value, the same value as that for DSL, is therefore adopted for HDSL.

12.3.3 *Measures to Handle the Slight Data Rate Difference in Opposite Directions*

It has been reported that at least for some private point-to-point DS1 circuits, it is not necessary that the line have network timing or that the Customer Premises Equipment (CPE) at the ends be loop-timed.[80] For instance, a customer can lease a DS1 line between two locations and attach his own digital channel banks equipped with voice channel units at the ends. The channel bank transmitter at each end will control the receiver frequency at the far end.

If only voice services are transported, neither channel bank needs to be loop-timed to the other. The free-running clock frequency accuracy requirement is between 50 bps (32 parts/million) and 200 bps (130 parts/million) for DS1 line output. As a result, the DS1 line frequency for such non-loop-timed circuits might be offset from each other by as much as 400 bps.

Another possible application for HDSL is to interconnect local area networks (LANs) through gateway or bridge devices. IEEE 802 LAN requirements for frequency accuracy and stability are less stringent than DS1 requirements. IEEE 802.3 for CSMA/CD LANs (such as ethernet) and IEEE 802.4 for token bus LANs require origination stations to have 100 parts/million frequency accuracy and repeating stations on 150 parts/million.

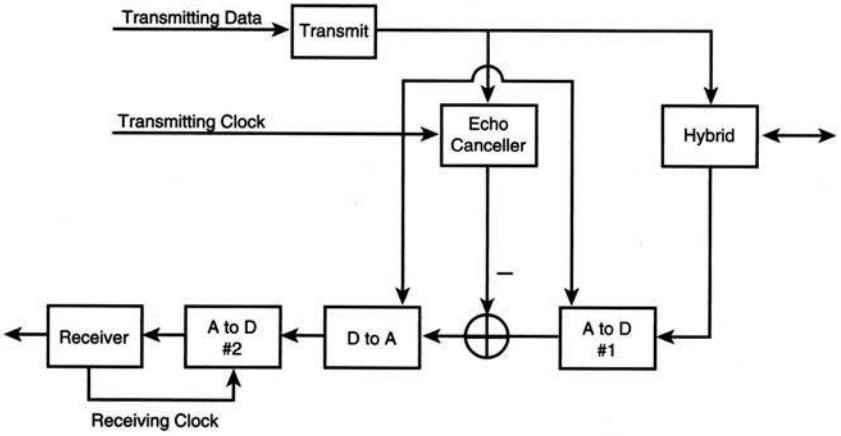
At DS1 rates, these requirements translate to 154 bps at best and 232 bps at worst. In the case that the loop-timing is not implemented, the throughput difference between each direction could be as much as 434 bps. In addition, there might also be a synchronization problem between a private digital network and a local exchange carrier.[81]

The asynchronous throughput in each direction poses a problem for the adaptive echo cancellation technique, which requires certain stability between transmitted and received data. While there have been some phase shifting insensitive echo cancellation techniques proposed in the past,[82,83] their performance under the dual-duplex HDSL asynchronous environment requires further study. In particular, the problem of missing bits or the occurrence of extra random bits should be addressed.

The same asynchronous problem has been in existence for the voice band modem for some time. A simple analog solution[84] is to perform ADC first for echo cancellation on the transmitting clock. The echo-removed signal is converted back to analog form with a Digital to Analog Converter. The echo-removed analog signal is then converted to digital for the equalizer using the recovered receiver clock. The extra DAC and ADC operation can be saved by using a digital technique,[85] especially in the form of a Finite Impulse Response (FIR) digital filter.[86]

Both the analog double A to D conversion and the digital FIR filter solutions are studied in the context of a dual-duplex HDSL application.[87] Figure 12.30 shows the structure of the double A to D conversion solution.

Figure 12.30 Double ADC Approach



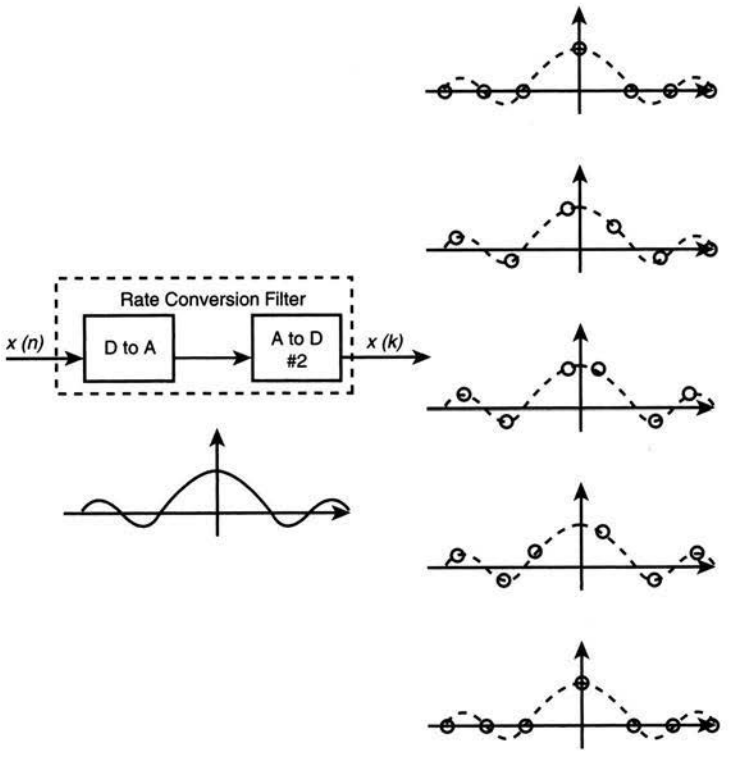
The required number of bits of the first A to D converter is related to the required echo cancellation level. The required number of bits of the second A to D converter is only related to the required signal-to-noise ratio of the DFE-based receiver. The echo cancellation level could be as high as 60 dB. In comparison, the required SNR corresponding to a 10^{-7} error rate for the 2B1Q signal is only about 30 dB. Hence, the cost of the second A to D converter is significantly less than the first one.

Rate Conversion Filters

To understand the operation of the rate conversion filter, let us first consider the case where the transmitting clock and the receiving clock have the same frequency but with a phase

delay of αT, for which $-0.5 < \alpha < 0.5$. We can use an FIR filter with Nα dependent filter coefficients to reconstruct the original signal in place of the combination of a D to A converter and an A to D converter. The output of the D to A converter has an ideal Nyquist pulse shape, and the output of the following A to D converter is then a sampled version of that Nyquist pulse (see Figure 12.31).

Figure 12.31 The D to A and A to D Equivalence of a Rate Conversion Filter



Possible digital Nyquist filters of different sampling phases are shown at the right side of Figure 12.31. We denote the FIR filter coefficient vector as

$$C(\alpha)^T = \left[c(\alpha)_0 \quad c(\alpha)_1 \quad \ldots \quad c(\alpha)_{n-1}\right]$$

To realize any $C(\alpha)$ for $-0.5 < \alpha < 0.5$, however, we need an infinite number of $C(\alpha)$s. On the other hand, because $C(\alpha)$ is a function of α, we can implement each $c(\alpha)_i$ as a polynomial function of α, that is, we let

$$c(\alpha)_i = c_{i,0} + c_{i,1}\alpha + c_{i,2}\alpha^2 + \ldots$$

Therefore, for a third-degree polynomial fitting of a length N FIR filter, we only need to store $4N$ values for any $C(\alpha)$, with this polynomial approach.

For the general asynchronous clock case, the assumption of a fixed delay between two clocks is not valid. The rate conversion FIR filter can still function well, however, if we let the filter coefficients be functions of instantaneous phase, $\alpha(k)$, that is

$$c(\alpha)^T = \left[c(\alpha(k))_0 \quad c(\alpha(k-1))_1 \quad c(\alpha(k-N+1))_{n-1}\right]$$

No additional computations are introduced if the filter output is generated according to

$$y_m = \sum_n x(n)c_{n,m}$$

$$z(k,m) = \sum \alpha(k)^m y_m$$

The index n corresponds to the transmitting clock, while k corresponds to the receiving clock. The $\alpha(k)$ is defined as

$$\alpha(k) = \mathrm{Re}\,m\left(\frac{nT_t - kT_r}{T_r}\right)$$

T_t is the period of the transmitting clock and T_r is of the receiving clock. If $\alpha(k)$ overflows, two new $x(n)$s are shifted into the delay line; if $\alpha(k)$ underflows, then no $x(n)$s are shifted in. Figure 12.32 shows the filter implementation.

This rate conversion filter can avoid the loss and the occurrence of extra bits. Because the length of the FIR filter and also the degree of polynomial are both limited, there are small errors introduced by this filter. In practice, for a sufficiently high degree of polynomial, $c_{i,j}$ are chosen offline to minimize the mean squared error according to

$$\int\int \left|\sum\sum \alpha^m C_{n,m} e^{jn\varpi T} - e^{j\varpi\alpha T}\right|^2 d\varpi d\alpha$$

Figure 12.32 The Structure of the Rate Conversion Filter

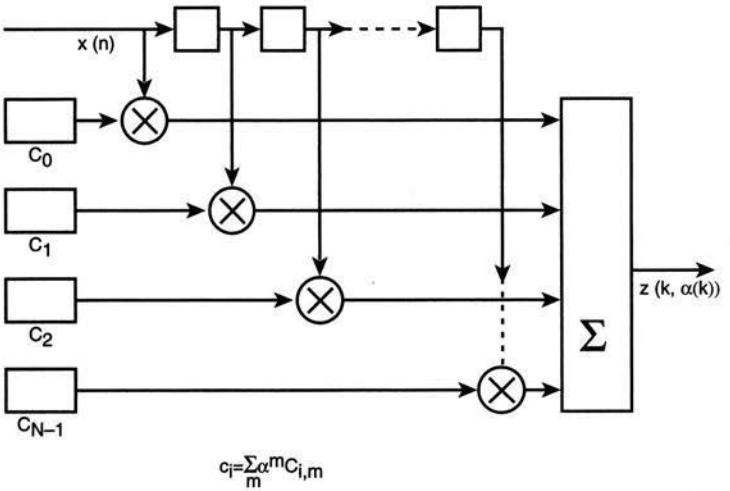


Table 12.5 shows a set of possible $c_{n,m}$ values corresponding to a baseband dual-duplex DSL with 50 percent excessive bandwidth and a sampling rate of 800 kilo-samples per second.

Table 12.5 A Possible Set of Rate Conversion Filter Coefficients

	m			
n	0	1	2	3
0	-0.005569	0.000732	0.022278	-0.00293
1	0.017471	-0.003174	-0.069885	0.012695
2	-0.040802	0.0103	0.163208	-0.041199
3	0.085114	-0.03157	-0.340454	0.126282
4	-0.183044	0.11792	0.732178	-0.47168
5	0.626038	-1.24646	-0.50412	0.98584
6	0.626038	1.24646	-0.50412	-0.98584
7	-0.183044	-0.11792	0.732178	0.47168
8	0.085114	0.03157	-0.340454	-0.126282
9	-0.040802	-0.0103	0.163208	0.041199
10	0.017471	0.003174	-0.069885	-0.012695
11	-0.005569	-0.000732	0.022278	0.00293

Variable Frame Approaches

The other approach to overcome the asynchronous clock problem is to introduce a variable frame structure. The clock difference between each direction can then be compensated

through *bit stuffing*. Bit stuffing is used to expand the length of a frame by inserting some dummy bits at some defined location. A method should be defined for the transmitter to decide when bit stuffing is required. The method should also allow the receiver to recognize stuffed bits to remove them. Two different variable frame formats have been proposed by Tellabs and PairGain[88] and Northern Telecom, Inc.[89]

Tellabs' and PairGain's Variable Frame Proposal

Tellabs' proposal starts with a 7-quats (or 14 bits, for each quat contains two bits) Barker code sync word. There are three bit-stuffing control bits evenly distributed between DS1 data blocks, indicating whether there is 0 or 2 quats stuffing at the end of an HDSL frame. The bit error rate is assumed to be $P_e = \eta$ for the following analysis.

A majority decision, two out of three, is made based on these three control bits. The probability of a wrong indication is as follows:

$$P_W = P_e(c_1)P_e(c_2) + P_e(c_1)P_e(c_3) + P_e(c_2)P_e(c_3) - 2P_e(c_1)P_e(c_2)P_e(c_3) = 3\eta^2 - 2\eta^3 = \eta^2(3-2\eta)$$

A sync word is considered recognized only when there is no error in any of its quats. The probability of receiving a bad sync word is as follows:

$$P_{bsw} = 1-(1-\eta)^{14}$$

A bad frame is declared when both the sync word and control bits are in error. The probability for a bad frame is as follows:

$$P_{w\&bsw} = P_W P_{bsw} = \eta^2\left(1-(1-\eta)^{14}\right)(3-2\eta)$$

For a frame repetition rate of 6 milliseconds, the expected bad frames per day is as follows:

$$E(Badframes/day) = P_{w\&bsw}\frac{24\times 3600}{6\times 10^{-3}}$$

Table 12.6 provides the performance estimation for the Tellabs variable frame structure proposal, assuming the preceding expressions are true. In Table 12.6, CFA stands for *Certainty of Frame Acquisition,* as will be examined shortly.

Table 12.6 Performance Estimation for Tellabs' Proposal

η	$1x10^{-3}$	$1x10^{-4}$	$1x10^{-5}$	$1x10^{-6}$	$1x10^{-7}$
P_w	$2.998x10^{-6}$	$3.000x10^{-8}$	$3.000x10^{-10}$	$3.000x10^{-12}$	$3.000x10^{-14}$
E(B.F./D.)	$6.005x10^{-3}$	$6.043x10^{-4}$	$6.048x10^{-5}$	$6.048x10^{-10}$	$6.048x10^{-13}$
CFA	95.51%	98.04%	98.19%	98.21%	98.21%

Because the sync word is only searched for at two places as suggested by the proposal, the framing will be hard to recover for two bad frames in a row. In this sense, the performance is more closely related to the expected number of bad frames than the probability of the loss of 6 sync words in a row as is commonly used in fixed frame formats.

The probability of misdetecting a sync word in n quats of random data is

$$P_{se(n-6)} = 1 - \left(1 - \frac{2}{4^7}\right)^{n-6}$$

The first sync word could occur equally likely between the first 7-quats position and the 2,346th 7-quats position. We have the probability of detecting a sync word for a frame length of N quats as

$$P_{se} = 1 - \left(1 - \frac{2}{4^7}\right)^{\frac{N-1}{2}}$$

using a mean random data length of

$$\frac{N-1}{2}$$

before the real sync word as the position where the frame search begins.

There are three cases in which the frame acquisition, two consecutive detections of two sync words, can be completed in three frames.[90] The first one is when there is no misdetection of a sync word before the successful detection of two consecutive sync words. The second case is for successful detection of two sync words after a misdetection of one sync word. The third case is when there is also no misdetection, but one missed real sync word before the successful detection of two sync words. Naming probabilities corresponding to these cases

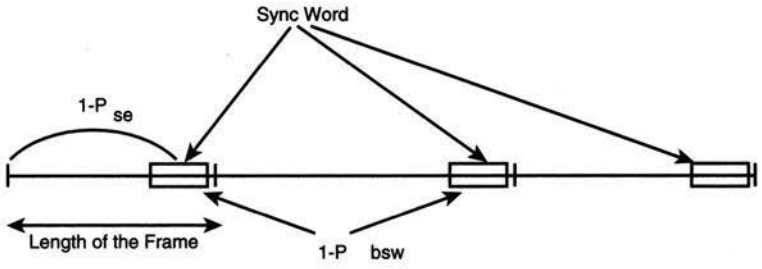
as P_1, P_2, and P_3, respectively, the probability of frame acquisition within the first three frames is

$$P_{3f} = P_1 + P_2 + P_3$$

We have the following (see Figure 12.33):

$$P_1 = (1 - P_{se})(1 - P_{bsw})(1 - P_{bsw})$$

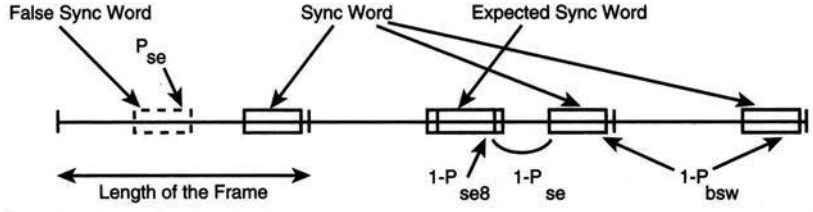
Figure 12.33 Calculation of P_1



We also have the following (see Figure 12.34):

$$P_2 = P_{se}(1 - P_{se2})(1 - P_{se})(1 - P_{bsw})(1 - P_{bsw})$$

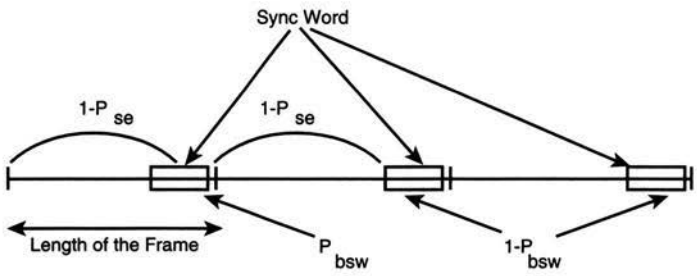
Figure 12.34 Calculation of P_2



Finally, we have the following (see Figure 12.35):

$$P_3 = (1 - P_{se})P_{bsw}(1 - P_{se})(1 - P_{bsw})(1 - P_{bsw})$$

Figure 12.35 Calculation of P_3



Given these probability calculations, we thus have

$$P_{3f} = (1-P_{bsw})^2(1-P_{se})(1+P_{se}(1-P_{se2})+(1-P_{se})P_{bsw})$$

The probability of frame acquisition in three frames is shown earlier in this section in Table 12.6 as CFA.

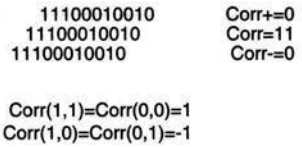
Although a 7-bit Barker code is used, the property of the Barker code is never utilized. In other words, the performance of the proposal will be exactly the same if the Barker code is replaced by any other bit pattern. Hence, through examining P_W, P_{bsw}, and $P_{w\&psw}$, the performance of the proposal depends primarily on P_W.

Northern Telecom's Variable Frame Proposal

Northern's proposal starts with an 11-bit Barker code sync word appearing as sign-only information in the first 11 quats of a frame. Stuffing occurs immediately prior to the sync word, in other words, at the end of an HDSL frame. The stuffing is a single quat plus or minus one quat. The deframer calculates correlation for three possible sync word windows. The window with the highest correlation is considered to contain the sync word. Considering the scenario for which the sync word is at the normal position, correlation for the advanced, normal, and delayed windows for zero bit error are 0, 11, and 0, respectively, assuming the following:

$$Corr(1,1) = Corr(0,0) = 1$$
$$Corr(1,0) = Corr(0,1) = -1$$
$$Corr(1,x) = Corr(0,x) = Corr(x,0) = Corr(x,0) = 0$$ (see Figure 12.36)

Figure 12.36 Calculation of P_E



Any bit error will reduce the correlation of the normal window by the amount equal to two times the number of errors. The correlation of the advanced or the delayed window will either be decreased or increased depending on error positions. The misdetection of the sync word position could happen only when the number of errors is larger than two. Hence, the probability of detecting a wrong sync word window is as follows:

$$P_E = \sum_{3}^{11} P(BitErrorNumber = i)\left[P(Corr+ > 11-i)\big|_c + P(Corr- > 11-i)\big|_c\right]$$

$$= \sum_{3}^{11} C_{11}^{i} P_e^{i}(1-P_e)^{11-i}\left[P(Corr+ > 11-i)\big|_c + P(Corr- > 11-i)\big|_c\right]$$

The values are as follows:

c means that the condition of $P_E = \sum_{3}^{11} P(BitErrorNumber = i)$ $P_e\eta$ is the bit error probability.

$Corr+$ means the correlation of the third window.

$Corr-$ means the correlation of the first window.

For

$$P_e \le 10^{-3}$$

we have

$$P_E \approx C_{11}^{3} P_e^{3}(1-P_e)^{8}\left[P(Corr+ > 5)\big|_c + P(Corr- > 5)\big|_c\right]$$

From Figure 12.36, there are five bit positions where an error will cause an increase in *Corr+* or *Corr–*. There are C_5^3 ways the 3-bit errors can be arranged in these five positions. There are C_{11}^3 ways the 3-bit error can be arranged in the entire sync word.

Therefore, we have

$$P(Corr+>5)\big|_c = P(Corr->5\big|_c) = \frac{C_5^3}{C_{11}^3} = \frac{10}{165}$$

Hence, we have

$$P_E \approx C_{11}^3 P_e^3 (1-P_e)^8 2\frac{C_5^3}{C_{11}^3} = 20\eta^3(1-\eta)^8$$

For a frame repetition rate of 6 milliseconds, the expected bad frames per day is

$$E(badframes/day) = P_E \frac{24\times 3600}{6\times 10^{-3}}$$

Table 12.7 lists the performance estimation, assuming the preceding expressions are true.

Table 12.7 Performance Estimation for Northern's Proposal

η	1×10^{-3}	1×10^{-4}	1×10^{-5}	1×10^{-6}	1×10^{-7}
P_E	1.984×10^{-8}	1.998×10^{-11}	2.000×10^{-14}	2.000×10^{-17}	2.000×10^{-20}
E(B.F./D.)	2.856×10^{-1}	2.878×10^{-4}	2.880×10^{-7}	6.048×10^{-10}	2.880×10^{-13}
CFA	52.25%	53.20%	53.29%	53.30%	53.30%

The detection of the first sync word is assumed to be a perfect match:

$$P_{ds} = (1-\eta)^{11}$$

The probability of misdetecting a sync word in n quats of random data is

$$P'_{se(n-10)} = 1-\left(1-\frac{2}{2^{11}}\right)^{n-10}$$

The probability of misdetecting a first sync word for a frame of length N is

$$P'_{se} = 1-\left(1-\frac{2}{2^{11}}\right)^{\frac{N-1}{2}}$$

The confirmation of the second sync word requires a perfect match in one of three possible windows. The probability of a match, assuming we know which window to look at, is P_{ds}. The probability of the sync word being in one of the three positions equals one. Therefore

$$P_{cs} = (1-\eta)^{11}$$

The possibility of misdetecting a second sync word has been omitted because we assume the framing circuitry knows where to "look" for the second sync word. Again, for three possible cases in which frame acquisition can be accomplished in the first three frames, we have

$$P_1 = (1 - P'_{se})P_{ds}P_{cs}$$

$$P_2 = P'_{se}(1 - P'_{se3})(1 - P'_{se})P_{ds}P_{cs}$$

$$P_3 = (1 - P'_{se})(1 - P'_{ds})(1 - P'_{se})P_{ds}P_{cs}$$

This leads to

$$P'_{3f} = P_{ds}P_{cs}(1 - P'_{se})(1 + P'_{se}(1 - P'_{se3}) + (1 - P'_{se})(1 - P_{ds}))$$

Assuming the preceding expressions, the performance estimation is also listed in Table 12.7 as CFA. The initial acquisition performance is much poorer than that of Tellabs' proposal due to usage of the magnitude bits for other purposes. As a result, Northern Telecom's proposal might not be considered adequate for the HDSL application.

Performance of Fixed Frame as a Reference

The performance of a traditional fixed frame format, where bad frames are declared for six consecutive bad sync words and an initial frame acquisition is declared after two perfect detections of the sync word, is also estimated for comparison. Both a 7-quat sync word and an 11-bit sync word are assumed for fixed frames. For the fixed frame, the probability of six consecutive bad sync words is as follows:

$$P_{sl} = \left(1 - (1-\eta)^{j}\right)^{6}$$

The probabilities for detection and confirmation of the sync word are both

$$P_d(1-\eta)^j$$

The probability of misdetecting a sync word in n quats of random data is

$$P_{s(n-j+1)} = 1-\left(1-\frac{2}{2^j}\right)^{(n-j+1)}$$

The probability of misdetecting a first sync word is

$$P_s = 1-\left(1-\frac{2}{2^j}\right)^{\frac{N-1}{2}}$$

The value j is 14 for the 7-quat sync word and 11 for the 11-bit sync word. The probability for frame acquisition in the first three frames is

$$P''_{3f} = (1-P_s)P_d^2 + P_s(1-P_{s1})(1-P_s)P_d^2 + (1-P_s)(1-P_d)(1-P_s)P_d^2$$

Table 12.8 lists the probabilities for the 7-quat sync word. Table 12.9 lists the probabilities for the 11-bit sync word. After achieving frame synchronization, the exact location of the sync word is known. The probability of detecting a valid sync word in 6 tries is very high. This phenomenon is known as *flywheel.*

Table 12.8 Performance Estimation for a 7-Quat Fixed Frame

η	$1x10^{-3}$	$1x10^{-4}$	$1x10^{-5}$	$1x10^{-6}$	$1x10^{-7}$
P_{sl}	$7.242x10^{-12}$	$7.500x10^{-18}$	$7.527x10^{-24}$	$7.529x10^{-30}$	$7.530x10^{-36}$
E(B.F./D.)	$6.258x10^{-4}$	$6.480x10^{-10}$	$6.504x10^{-16}$	$6.504x10^{-22}$	$6.504x10^{-28}$
CFA	95.51%	98.04%	98.19%	98.21%	98.21%

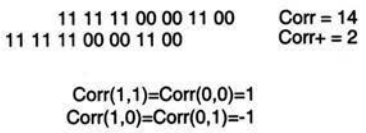
Table 12.9 Performance Estimation for an 11-Bit Fixed Frame

η	$1x10^{-3}$	$1x10^{-4}$	$1x10^{-5}$	$1x10^{-6}$	$1x10^{-7}$
P_{sl}	$1.719x10^{-12}$	$1.766x10^{-18}$	$1.771x10^{-24}$	$1.772x10^{-30}$	$1.772x10^{-36}$
E(B.F./D.)	$1.484x10^{-4}$	$1.526x10^{-10}$	$1.531x10^{-16}$	$1.531x10^{-22}$	$1.531x10^{-28}$
CFA	52.29%	53.24%	53.33%	53.34%	53.34%

The Double-Barker Code Variable Frame Approach

Analysis shows that while normal operation performances—expected bad frames per day—of both proposals are close to each other, their performance is not as good compared with that of a fixed frame format. On the other hand, the initial acquisition performance—certainty of frame acquisition in the first three frames—of the Tellabs proposal is much better than the Northern Telecom proposal. Based on Tellabs' and Northern Telecom's proposals, a double-coded 7-quat Barker code for the variable frame format is proposed (see Figure 12.37).[91]

Figure 12.37 Calculation of P_E



A two-window with autocorrelation detection scheme is used, with either 0 or 2 quats of bit stuffing at the end of each frame. The probability of detecting a wrong sync word window is as follows:

$$P_E = \frac{1}{16}\sum_{i=3}^{14} P(BitErrorNumber = i)P(Corr+ \geq 14 - 2i)\Big|_c$$

$$+\frac{10}{16}\sum_{i=4}^{14} P(BitErrorNunber = i)P(Corr+ \geq 14 - 2i)\Big|_c$$

$$\frac{1}{4}\sum_{i=5}^{14} P(BitErrorNumber = i)P(Corr+ \geq 14 - 2i)\Big|_c$$

$$\approx \frac{1}{16} C_{14}^3 P_e^3 (1 - P_e)^{11} P(Corr+ \geq 8)\Big|_c$$

for

$$P(Corr+ \geq 6)\Big|_c = \frac{C_6^3}{C_{14}^3} = \frac{20}{C_{14}^3}$$

and

$$P_e = \eta$$

As a result, we have

$$P_E \approx \frac{5}{4}\eta^3(1-\eta)^{11}$$

Table 12.10 summarizes the performance estimation for a 7-quat double-Barker code.

Table 12.10 Performance Estimation for a 7-Quat Double-Barker Code

η	$1x10^{-3}$	$1x10^{-4}$	$1x10^{-5}$	$1x10^{-6}$	$1x10^{-7}$
P_E	$1.236x10^{-9}$	$1.249x10^{-12}$	$1.250x10^{-15}$	$1.250x10^{-18}$	$1.250x10^{-21}$
E(B.F./D.)	$1.780x10^{-2}$	$1.780x10^{-5}$	$1.780x10^{-8}$	$1.800x10^{-11}$	$1.800x10^{-14}$
CFA	95.51%	98.04%	98.19%	98.21%	98.21%

The certainty of frame acquisition in the first three frames is the same as the Tellabs proposal. The 7-quat double-Barker code can be coded as [+3 +3 +3 −1 −1 +3 −1] following the conventional gray code or as [+3 +3 +3 −3 −3 +3 −3] in conjunction with a special autocorrelation decoder. Table 12.11 summarizes the performance of using an 11-quat double-Barker code, which is also estimated according to similar expressions.

Table 12.11 Performance Estimation for an 11-Quat Double-Barker Code

η	$1x10^{-3}$	$1x10^{-4}$	$1x10^{-5}$	$1x10^{-6}$	$1x10^{-7}$
P_E	$1.548x10^{-14}$	$1.572x10^{-19}$	$1.575x10^{-24}$	$1.575x10^{-29}$	$1.575x10^{-34}$
E(B.F./D.)	$2.229x10^{-7}$	$2.264x10^{-12}$	$2.268x10^{-17}$	$2.268x10^{-22}$	$2.268x10^{-27}$
CFA	97.77%	99.77%	99.98%	100.00%	100.00%

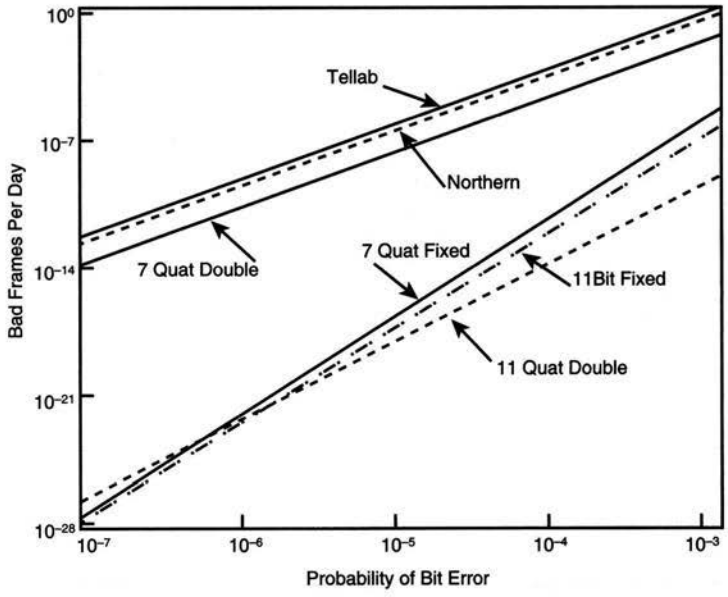
Figure 12.38 illustrates the performance comparison (in expected bad frames per day) between a 7-quat double-Barker code and an 11-quat double-Barker code.

Figure 12.39 illustrates the performance comparison (in initial frame acquisition) between a 7-quat double-Barker code and an 11-quat double-Barker code.

The proposed double-Barker code variable frame format has performance closer to that of a fixed frame format. The 7-quat double-Barker code variable frame format requires three less bits compared with Tellabs' proposal and needs an autocorrelation operation only 3 bits

longer than Northern's proposal. The 7-quat double-Barker code frame format is therefore adopted for the HDSL variable frame application. The performance of this double-Barker code variable frame format is also very robust under impulse noise.[92]

Figure 12.38 Expected Bad Frames per Day

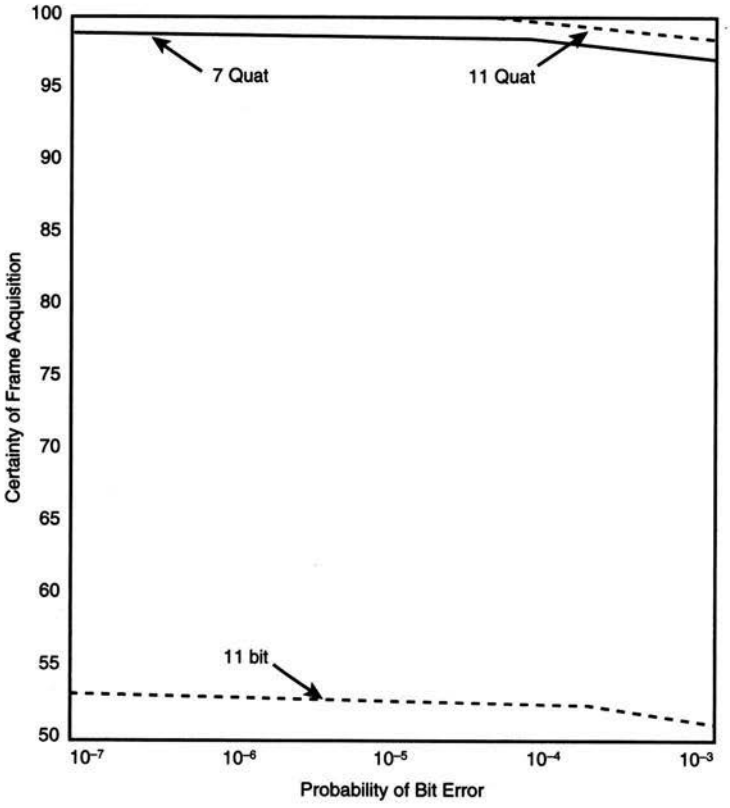


12.4 *HDSL Network Interfaces*

The dual-duplex HDSL is designed to be a cost-effective repeaterless T1 transceiver. A T1 transceiver is generally used to carry the DS1 traffic for the Public Switch Telephone Network (PSTN). The DS1, which stands for Digital Signal 1, is the first level of the North American digital hierarchy with a nominal rate of 1.544 Mbps. A specific implementation of DS1 is further referred to as DSX-1. ANSI T1.107[93] defines the DS1 bit rate and format.

ANSI T1.102[94] defines the DS1 electrical interconnection requirements for frequency accuracy and frequency stability, line code, impedance, and pulse shape for DSX-1 cross-connect points. ANSI T1.403[95] and T1.408[96] define the requirements for DS1 signals for network-customer interfaces.

Figure 12.39 Initial Certainty of Frame Acquisition



12.4.1 *The DS1 Infrastructure*

For most network applications, the DS1 signal has a 125 μsec frame structure (see T1.107) with 192 payload bits and one framing/overhead bit. The first bit is the DS1 frame bit, called the *F* bit. The F frame channel is composed entirely of framing information, as in the Superframe format, or made up of a framing pattern, a data link, and a 6-bit cyclic redundancy check code, as in the Extended Superframe format. The framed DS1 payload is thus 1536 kbps (192 bits/125 μsec) with an overhead of 8 kbps (1 bit/125 μsec).

Most applications define the 192 bits in a frame as 24 8-bit DS0 channels of 64 kbps each. These 8-bit time slots are called bytes. These DS0 channels traditionally carried digitally encoded telephony voice signals, but they can be any 64 kbps signal.

The DSX-1 signal is a bipolar signal where binary one levels alternate in polarity and have a peak magnitude ranging from 2.4 to 3.6 volts. A zero binary level is represented by zero volts. Time wave shape templates are specified for the binary one pulses. This bipolar signal line code is also called *Alternate Mark Inversion* (AMI). A variation on the bipolar line code is *bipolar with 8-zero substitution* (B8ZS). A B8ZS transmitter replaces strings of eight consecutive zeros by a specific eight-symbol pattern that violates the alternating polarity rule.

The receiver detects the pattern and restores the original eight-zero-bit string. The B8ZS encoding ensures sufficient pulse density for older line regenerators to derive timing from the incoming signal. At the network side, the DSX-1 wiring is connected by wire wrap pins on the back plane of the frame containing the device. At the customer side, the DSX-1 wiring is connected via a standard RJ-48C jack and also by screw-down strips.

A T1 transceiver is a transmission system designed to transport bi-directional 1.544 Mbps signals over twisted-pair loops. Each direction is transmitted on a separate twisted-pair loop with a transmitter at one end and a receiver at the other end. Line repeaters are placed in the outside plant at intervals, typically between 3 and 6 kft, determined by the wire gauge and the number of other T1 systems sharing the cable. Bridged taps are not allowed. They are removed during the T1 line installation from conventional telephone subscriber loops.

With the use of many repeaters, a range of 100 miles is possible. While DS1 signals have many options for framing or nonframing, and many possibilities exist for multiplexing the DS1 payload, HDSL is intended to be independent of the details beyond the 1.544 Mbps and the +–3 volt bipolar line encoding, just as a T1 line repeater is.

HDSL DS1 Interface

Dual-duplex HDSL systems could have transported DS1 signals by multiplexing the signal onto the two HDSL transceiver-loop sub-systems on a bit-by-bit, a byte-by-byte, or a group-by-group basis or other schemes. One concern is to minimize end-to-end transmission delay. Another consideration is possibly taking advantage of the DS1 service transport with only one loop operation.

A bit-by-bit mapping would make the remaining payload virtually unusable. The Group 1 and Group 2 multiplexing scheme proposed in the standards committee report appears to be a reasonable compromise solution, adding about 63 μsec (half a DS1 frame) more delay than the minimum, but providing contiguous DS0s in one loop operation.

For transport via a dual-duplex HDSL system, the DS1 signals are disassembled into three components:

- The DS1 framing (F) bits (if present)
- The Group 1 bytes (1 through 12)
- The Group 2 bytes (13 through 24)

The DS1 framing bit occurs at every 0-th bit position in each 125 μsec DS1 signal frame at a nominal bandwidth of 8 kbps. A DS0 time slot is made up of eight bits (one byte) in each 125-microsecond frame and has a nominal bandwidth of 64 kbps. Specifically, it was defined that the DS1 framing F bits, if present, shall be transported via both HDSL transceivers over two loops and that the HDSL system shall have a default option to detect B8ZS line encoding on the incoming DS1 bitstream and recreate B8ZS on the outgoing DS1 bitstream.

DS1 components of the HDSL payload for each transceiver require 12×64=768 kbps nominal bandwidth. The 8 kbps of DS1 F bit overhead will be added to both transceiver pairs. The composite HDSL payload through each transceiver and on each loop is thus a nominal 768+8=776 kbps. For an unframed DS1 signal, the HDSL shall choose a bit to represent the F bit position and count bytes relative to the chosen bit to represent the Group 1 and Group 2 bytes.

The dual-duplex HDSL system should accommodate differences in total length (working length plus the sum of bridged tap lengths) between two loops as large as 1 kilometer (3.3 kft). The HDSL system also should accommodate DS1 signals when the two directions have frequency offsets relative to each other or to the network or to the HDSL timing averaging as much as +–130 PPM.

Figure 12.40 shows the detailed DS1-to-HDSL-to-DS1 signal conversion process.

Figure 12.41 shows the HDSL framing structure during the normal operation after the initialization.

The HDSL frame starts with a seven-symbol synchronization word followed by 1 quat of other HDSL overhead. Here, a quat consists of two bits and occupies one symbol interval. The synchronization word is a 7-quat Barker code (+3 +3 +3 –3 –3 +3 –3) in one direction and the time reverse (–3 +3 –3 –3 +3 +3 +3) of the same Barker code in the other direction. This double assignment of the code word binary value is referred to as a *double-Barker code.*

The double-Barker code has been shown to have superior autocorrelation properties in the presence of noise. To further enhance the SNR in the initialization process, the double-Barker code is coded with 11 for +3 and 00 for –3 instead of the gray code used during the normal transmission.

Figure 12.40 The HDSL-DS1 Payload Interface

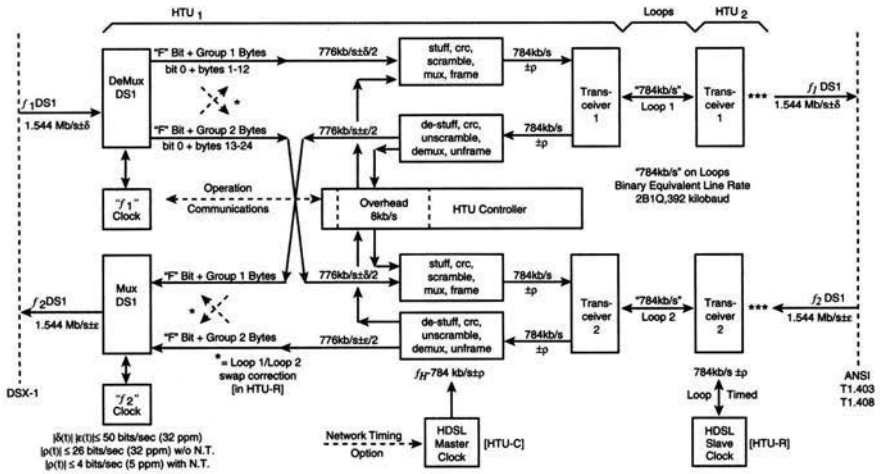
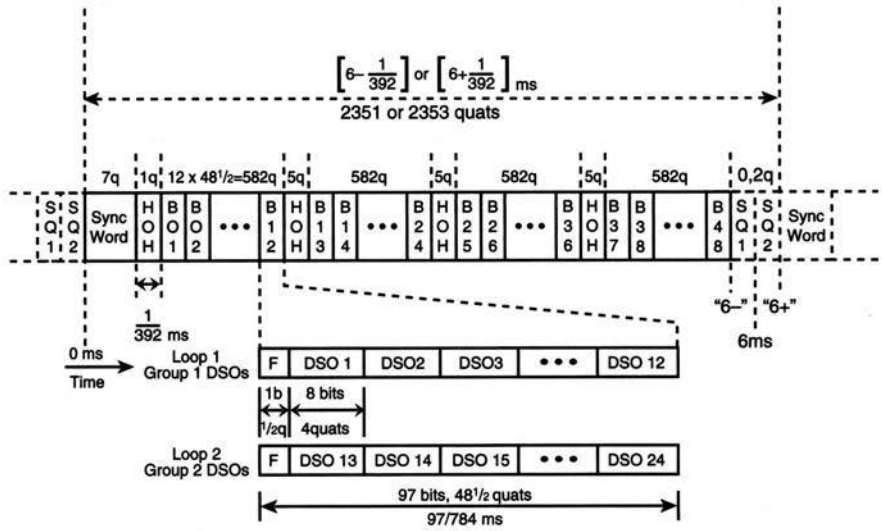


Figure 12.41 HDSL Frame Structure



HDSL DS1 Payload Block (48 per HDSL Frame): Group 1 & Group 2 Split

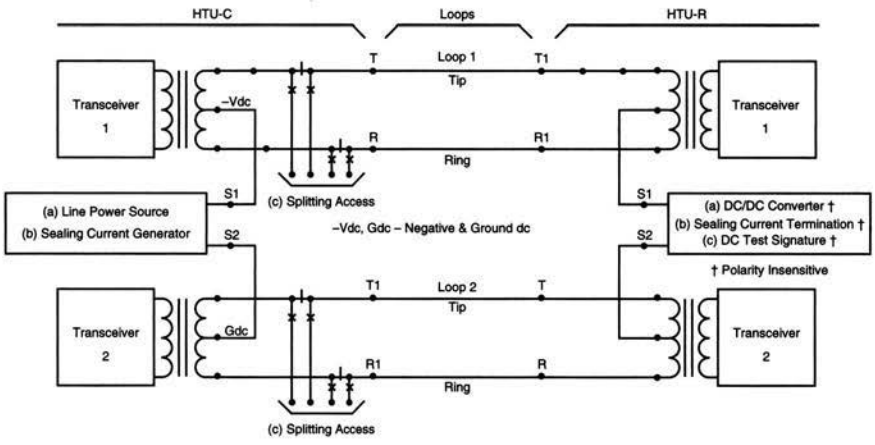
After the overhead quat, there are four groups of HDSL payloads. Each of the first three HDSL payload groups has 581 payload quats and five overhead quats. The last HDSL payload group has only 581 payload quats. At the end of the HDSL frame there could be two stuffing quats. The total length of an HDSL frame is either 2,351 or 2,353 quats corresponding to 6–1/392 ms or 6+1/392 ms. Each HDSL payload group is further divided into 12 DS1 blocks. Each DS1 block is composed of a DS1 framing bit, F (if present, or an HDSL-chosen substitute), and 12 bytes of DS0 data, for a total of 97 bits encoded to 481/2 quats.

Used as a dual-duplex T1 transceiver, each HDSL frame covers four DS1 frames. With a total HDSL overhead of 24 quats or 48 bits plus 24 duplicated F bits per frame, the HDSL binary equivalent line rate is 784 kbps for a single pair or 1.568 Mbps for two pairs.

HDSL Line Powering Options

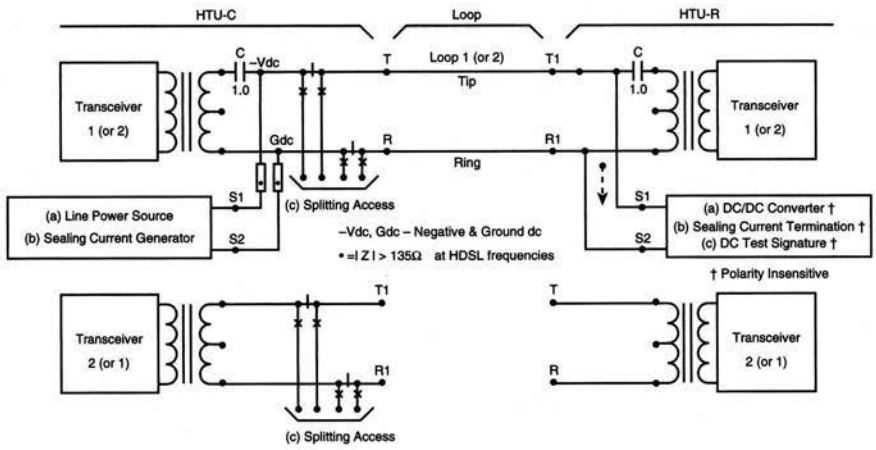
An HDSL Terminal Unit-Remote (HTU-R) can be powered through a central office, as indicated by Figure 12.42.

Figure 12.42 Dual-Loop HDSL Loop Powering



One pair is used as ground while the other pair is used to supply the necessary voltage. The HTU-R should have a DC-DC converter to re-generate a proper and stable voltage level suitable for the HDSL transceiver. The HTU-R can also be powered through a single loop, as indicated by Figure 12.43.

Figure 12.43 Single-Loop HDSL Loop Powering



In the single-loop case, capacitors need to be inserted between line transformers and the loop to prevent the short circuit of the power supply.

12.5 HDSL Technical Report

The draft T1E1.4 HDSL technical report[97] addresses interface details for the 2B1Q line code as well as for the alternative CAP HDSL line code. Specifics for the 2B1Q line code covered are pulse shaped and transmit power spectrum, test loops, test procedures, frame structure, scrambling method, start-up and training, and jitter requirement, as well as Operation, Administration, Maintenance (OAM), and provisioning and Embedded Operations Channels (EOC). Specifics for the CAP line code covered are transmit power spectrum, scrambling method, frame structure, and start-up procedure.

12.5.1 2B1Q Bit-to-Symbol Encoding

The 2B1Q line code uses the gray code to encode every two bits into a quaternary symbol, as shown in Table 12.12.

Table 12.12 The HDSL Gray Encode

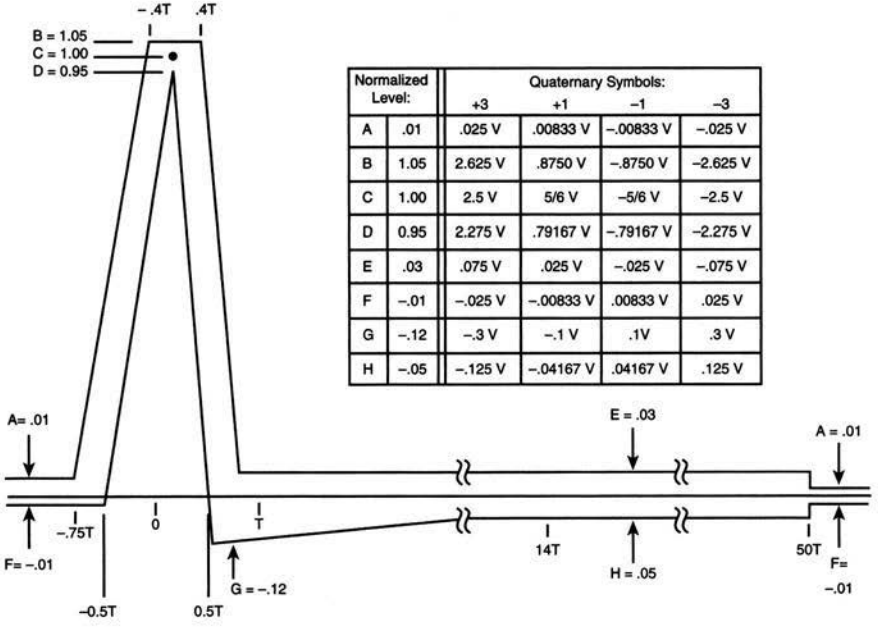
First Bit (Sign)	Second Bit (Magnitude)	Quaternary Symbol (Quat)
1	0	+3
1	1	+1

First Bit (Sign)	Second Bit (Magnitude)	Quaternary Symbol (Quat)
0	1	–1
0	0	–3

12.5.2 *Transmit Signal Pulse Mask*

The HDSL line signal from a transmitter should confirm the pulse mask defined in the time domain (see Figure 12.44).

Figure 12.44 HDSL Transmit Signal Pulse Mask



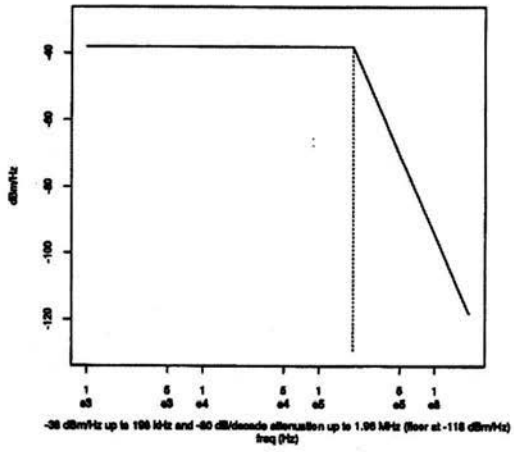
Normalized Level:		Quaternary Symbols: +3	+1	–1	–3
A	.01	.025 V	.00833 V	–.00833 V	–.025 V
B	1.05	2.625 V	.8750 V	–.8750 V	–2.625 V
C	1.00	2.5 V	5/6 V	–5/6 V	–2.5 V
D	0.95	2.275 V	.79167 V	–.79167 V	–2.275 V
E	.03	.075 V	.025 V	–.025 V	–.075 V
F	–.01	–.025 V	–.00833 V	.00833 V	.025 V
G	–.12	–.3 V	–.1 V	.1V	.3 V
H	–.05	–.125 V	–.04167 V	.04167 V	.125 V

The compliance of the transmit signal in the time domain can be achieved by choosing the proper combination of a transmitter filter and a hybrid circuit. The transmit pulse shape can be measured in real time using either a single pulse or a known pulse sequence with some signal processing.

12.5.3 Transmit Signal Power Mask

Figure 12.45 shows the upper bound of the average power spectral density of the transmit signal, which can be summarized as that the power density should be –38 dBm/Hz from 0 Hz to 196 kHz, –80 dB/decade attenuation from –38 dBm/Hz at 196 kHz to –118 dBm/Hz at 1.96 MHz, and –118 dBm/Hz above 1.96 MHz.

Figure 12.45 HDSL Transmit Signal Power Mask

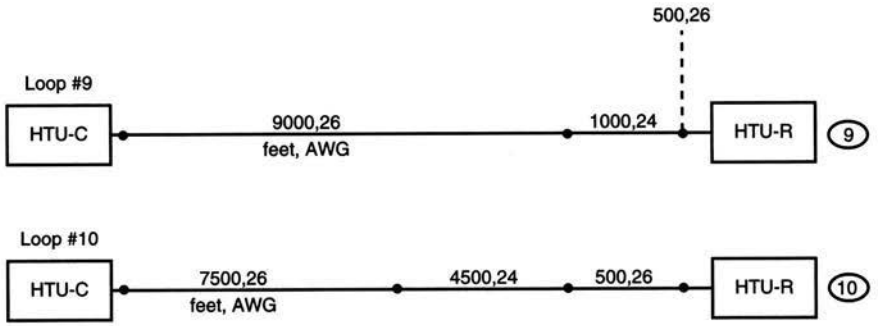


The –80 dB/decade attenuation slope can be obtained with a fourth-order Chebychev low-pass transmit filter. The total power of the transmit signal is between 13 dBm and 14 dBm over the frequency band between 0 Hz to 784 kHz. The power consumption of the line driver can be made less than 100 mW for using a 5-volt power supply. The worst case power consumption on the loop is $2.5^2/135 = 46.3$ mW. Therefore, the power consumption of a line driver can be 92.6 mW for an circuit efficiency of 50 percent.

12.5.4 HDSL Test Loops

In addition to the eight CSA loops used for the simulation study, a zero-length loop and two extended (longer than) CSA loops also are used for the HDSL transceiver test. They are named CSA test loop –0, –9, and –10, respectively, for a total of 11 CSA test loops. The 49-disturber 1 percent worst-case self-NEXT is the test noise environment. A performance margin of at, or better than, 6 dB at an error rate of 10^{-7} is expected for these eight CSA test loops. No margin objectives are provided for these two extended CSA loops (see Figure 12.46).

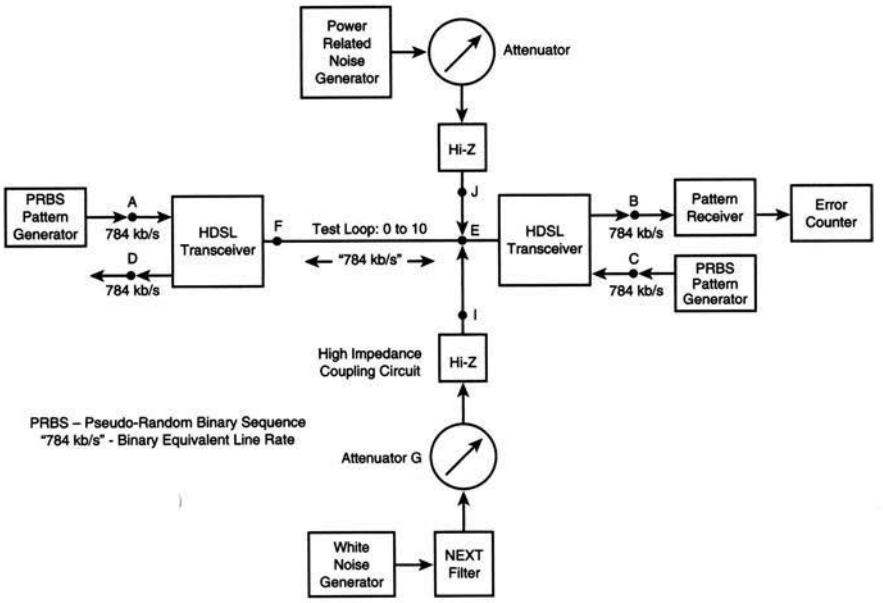
Figure 12.46 Additional Two CSA Test Loops



12.5.5 *HDSL Performance Test Configuration*

Figure 12.47 shows a general HDSL performance test configuration.

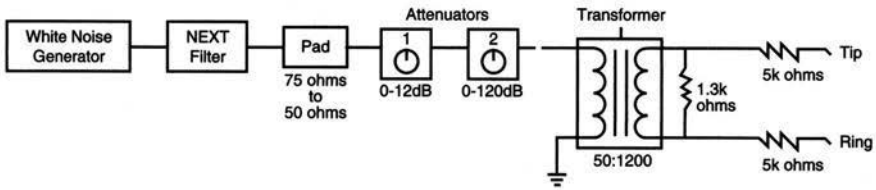
Figure 12.47 The Test Environment



The HDSL transceivers are made active in both directions, while the bit error rate of only one direction is measured at a time. Real loops or a loop emulator can be used for these test

loops. Simulated NEXT noise as well as impulse noise wave forms are applied to the receiver front end through high-impedance coupling circuits (see Figure 12.48). During the NEXT performance margin test, impulse noise wave forms are not injected.

Figure 12.48 An Implementation of the NEXT Noise Coupling Circuit



A particular implementation[98] of the NEXT noise emulation circuit has a two-stage adjustable attenuator, a 1:24 ratio transformer, and a coupling impedance of about 10 kilo-ohms. The NEXT filter has a transfer function, which is the combination of the 2B1Q transmit signal spectrum, the fourth-order Chebychev transmit lowpass filter, and the simplified 49-disturber 1 percent worst-case NEXT coupling function.

Figure 12.49 shows the transfer function of a possible NEXT filter. Due to the limited power of the white noise generator, the net loss of this NEXT filter should be limited to less than 15 dB.[99]

Figure 12.49 Transfer Function of a NEXT Filter

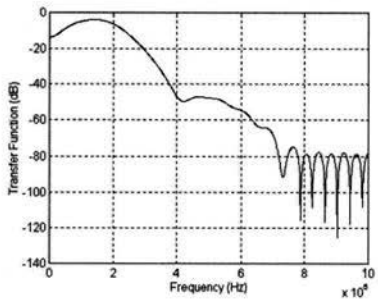
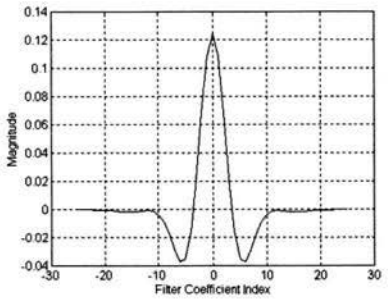


Figure 12.50 shows the impulse response of this NEXT filter.

Figure 12.50 Impulse Response of the NEXT Filter



12.5.6 *Measurement Calibration Procedures*

The transmit signal power, the emulated NEXT noise power, and the relative strength between the signal and noise power all need to be measured and calibrated before a meaningful error rate measurement can be performed. The transmit power spectral density (PSD) is first measured. The transmit PSD is then multiplied by the transfer function of the simplified 49-disturber 1 percent worst-case crosstalk loss model to obtain the PSD of the expected NEXT noise. The total expected NEXT power is computed by integrating the PSD of the NEXT noise.

The total power of the emulated NEXT measured at the output of a high-impedance coupling circuit, terminated in half of the nominal loop impedance, is adjusted (by adding attenuation) until the power is 3 dB higher than the total power calculated for the expected NEXT noise. This power level of the emulated NEXT is the *reference level* (also called the *point of 0 dB margin*). The shape of the PSD for the emulated NEXT measured at the output of the coupling circuit is also compared with the shape of the calculated PSD of the expected NEXT to certify the correctness of the NEXT filter. The difference between the two curves should be less than 2 dB at any frequency.

To find the performance margin level at an error rate of 10^{-7}, the attenuation of the NEXT power is removed repeatedly in 1 dB steps, and the errors are counted for at least two minutes at each attenuation setting until a total of 10 errors are counted in any two-minute interval. The corresponding error rate is

$$\frac{10}{2\times 60\times \dfrac{1.544\times 10^{6}}{2}} = \frac{10\times 10^{-7}}{6\times 1.544} \approx 1\times 10^{-7}$$

The two-minute observation interval is necessary to have a high enough confidence level on the determination of the error rate. For an error rate of η, the probability of observing an error at a particular independent sampling point is η, while the probability of not observing an error at a particular sampling point is $1-\eta$. Therefore, the probability of not observing any error for n independent sampling points is

$$\bar{\gamma}_n = (1-\eta)^n$$

Similarly, the probability of observing any error for n independent sampling points is

$$\gamma_n = 1-(1-\eta)^n$$

For $\eta = 1\times10^{-7}$, we have γ_n calculated in Table 12.13 for different values of window size n.

Table 12.13 The Probabilities of Window Sizes

n	$\bar{\gamma}_n = (1-\eta)^n$	$\gamma_n = 1-(1-\eta)^n$
1×10^7	0.3678794229711	0.6321205770289
2×10^7	0.1353352698456	0.8646647301544
3×10^7	0.04978706097842	0.9502129390216
5×10^7	0.006737945332332	0.9932620546677
1×10^8	4.539990730149e-5	0.9999546000927

According to results in Table 12.13, a choice of selecting a window size to capture only one error will have only a 63-percent chance of observing any error, while a choice of selecting a window size to capture 10 errors will have a 99.99-percent chance of observing errors. The window of capturing 10 errors can also be viewed as averaging a few windows with higher than 95-percent chance of capturing errors. Therefore, with the absence of the error occurrence statistics, a window size of capturing 10 errors is considered adequate.

12.5.7 Scambler and Descrambler

All data bits, except the 14 bits for the double-Barker code and stuff quart bits, is scrambled with a 23rd-order polynomial prior to the 2B1Q encoder. In the upstream direction, the polynomial is

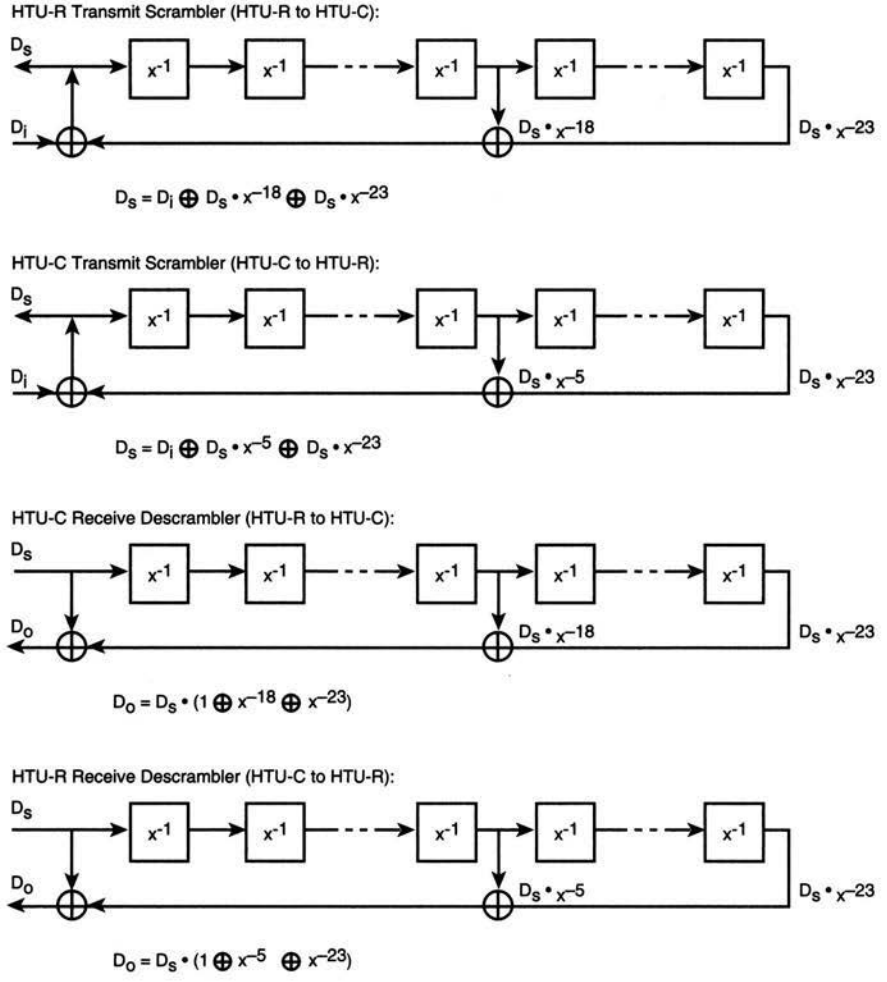
$$x^{-23} \oplus x^{-18} \oplus 1$$

In the downstream direction, the polynomial is

$$x^{-23} \oplus x^{-5} \oplus 1$$

The symbol $\oplus$ denotes a modular 2 addition. The scramblers and descramblers are self synchronized, as shown in Figure 12.51.

Figure 12.51 Scrambler and Descrambler



12.5.8 HDSL Initialization Procedure

At installation, following power failures or after temporarily disconnecting an HTU or switching off its power, the HDSL system must synchronize transmission between the two ends. In these situations, the HDSL system utilizes a specific startup sequence in order to achieve proper operation. Also, in the event that synchronization is lost for an extended period, it may be necessary to retrain the transceivers to restore proper operation.

States

There are seven states defined for the operation of both HDSL Terminal Unit-Central Office (HTU-C) and HTU-R:

- *Inactive.* The HTU-C/R is inactive.
- *Activating.* The HTU-C/R is becoming active.
- *Active-Tx.* The HTU-C/R is transmitting.
- *Active-Rx.* The HTU-C/R is receiving.
- *Active-Tx/Rx.* The HTU-C/R is transmitting and receiving.
- *Pending De-Activation.* The HTU-C/R is becoming deactivated.
- *De-Activated.* The HTU-C/R is deactivated.

Line Signals

Two level signals, RS0 and CS0, are used in the early stage of the startup procedure. Four level signals, RS1 and CS1, are used in the late stage of the startup procedure. Two level signals, RS0 and CS0, are derived from the scrambling of a bit stream with all 1s. Only the sign of the output quat is used to determine whether the transmit symbol should be +3 or –3.

Synchronization words are periodically inserted into RS0 and CS0. RS0 is used by the HTU-R transmitter and CS0 is used by the HTU-C transmitter. The four level signals, RS1 and CS1, include normal framing synchronization words. The overhead channel and stuffing quats are all included in RS1 and CS1. A portion of the HDSL payload is filled with scrambled 1s. RS1 is used by the HTU-R transmitter, and CS1 is used by the HTU-C transmitter.

Indication Bits and Registers

One indication bit, bit #3536, is used to indicate that a receiver is ready for receiving data. The indication bit is used by the HTU-R as INDR and by the HTU-C as INDC. The INDR is set to 1 when the HTU-R receiver is ready, and the INDC is set to 1 when the HTU-C receiver is ready.

There are two registers, LOS and LOST, used for both HTU-C and HTU-R for the purpose of received signal detection. The LOS register is set to 1 when the received signal is not detected and set to 0 when the received signal is detected. The LOST register is set to 1 when LOS=1 for 1 second. The LOST register is set to 0 when the received signal is detected.

There are two registers, LOSW and LOSWT; two major states, OUT OF SYNC and IN SYNC; and six transition states used for both HTU-C and HTU-R for the purpose of tracking the detection of the synchronization word, Double Barker Code. The LOSWT register is set to 1 when a synchronization word is missed during a period. The LOSWT register is set to 0 when a synchronization word is detected during a period. The LOSW register is set to 1 when synchronization is lost, which is denoted by the OUT OF SYNC state. The LOSW register is set to 0 when synchronization is regained, which is denoted by the IN SYNC state.

From the OUT OF SYNC state, it takes two consecutive successful detections of the synchronization word to get into the IN SYNC state. The state after the first successful detection of a synchronization word is state 0. From the IN SYNC state, it takes six consecutive losses of the synchronization word to get into the OUT OF SYNC state. There are state 1, state 2, state 3, state 4, and state 5 after the first, second, third, fourth, and fifth losses of the synchronization word, respectively.

Startup Sequence

When powered on, the HTU-C is initially in the Inactive State. If the activation request signal ACTREQ is set to 1, which is normally the default when powering on, the HTU-C proceeds immediately to the Activating State. In the Activating State, the HTU-C executes the startup procedure.

The startup procedure covers the Activating State, Active-Tx State, and Active-Rx State. Initially, the startup procedure is started after both units have been powered on and any self-tests completed. When the HTU-C starts the startup procedure, it automatically begins to

transmit. When the HTU-R starts the startup procedure, it automatically remains silent and waits to detect signal power from the HTU-C. The 30-second Activation Timers at the HTUs start at the beginning of the startup procedure.

At the startup, the HTU-C begins to transmit a two-level signal, CS0. During the first T2 seconds (2.0 +/–0.1 seconds), the HTU-R remains silent. At the end of this time the HTU-R begins to transmit a two-level signal, RS0. Within T4=4 seconds of beginning transmission, the timing recovery loop within the HTU-R should have reached a stable timing phase. Since the HTU-R is loop timed, this implies that the HTU-R transmitter has reached a stable transmit phase at this point as well.

The HTU-C will switch from two-level transmission to four-level transmission T1 seconds after the end of T2, providing that the HTU-C has detected signal power from the HTU-R at a point T2 seconds from the beginning of the startup procedure. The value of T1 can range from 5 to 10 seconds. If no signal power is detected from the HTU-R at the end of T2, the HTU-C will continue in CS0 until the 30-second Activation Timer forces it to leave the Activating State. If signal power is detected from the HTU-R after T2 seconds but before the 30-second Activation Timer expires, the HTU-C will reset the 30-second Activation Timer at this occurrence for a point two seconds into the startup procedure to correspond to the point at the end of T2 in the startup procedure.

Within T3 seconds from the time the HTU-C changes from CS0 to CS1 and the four-level signal is detected at the HTIJ-R, the HTU-R must switch to a four-level signal, RS1. The T3 interval can range from zero to four seconds; the precise duration is under control of the HTU-R.

Within 30 seconds from the start of the timeline, the HTUs should be transporting normal 2BIQ data. With a successful startup, the switch to normal data is accomplished after frame synchronization is acquired—either by signaling via indicator bits in the overhead channel or by the expiration of the 30-second Activation Timer.

If the activation process fails, then the Activation Timer expires 30 seconds after the start of the timeline and the unit moves to the De-Activated State.

After entering the De-Activated State, either due to a period of synchronization word loss (LOSWT=1) or due to expiration of the 30-second Activation Timer, the HTU-C transmitter goes silent and the HTU-C begins to look for a loss of signal power from the HTU-R. When LOS=1 for one second, the timer LOST is set to 1 and the unit moves to the Inactive State. If (or when) ACTREQ=1, the unit will proceed once again to the Activating State, and another attempt is made at the startup process.

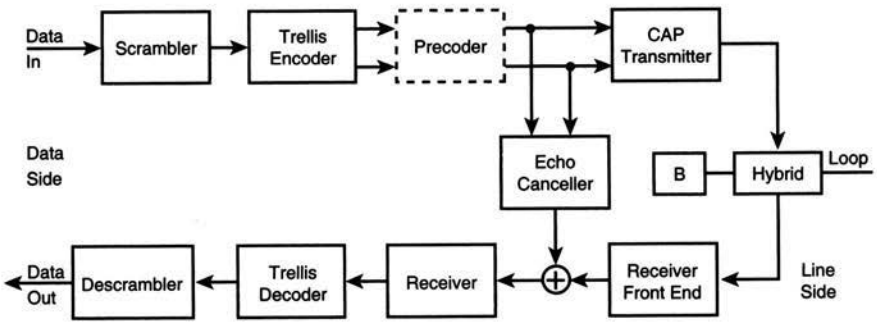
After entering the De-Activated State, the HTU-R transmitter goes silent and the HTU-R begins to look for a loss of signal power from the HTU-C. When a loss of signal power is detected (LOS=1), the unit moves immediately to the Inactive State. When signal power is detected from the HTU-C (LOS=0), the unit will proceed once again to the Activating State, and another attempt is made at the startup process.

An HTU receiving a signal to go to QUIET mode will make the transition to the De-Activated State. The signal for QUIET mode can be generated from a network command, the front panel, or by some other means.

12.5.9 *Key Features of the Optional CAP HDSL*

Figure 12.52 shows a functional block diagram of an HDSL transceiver using the optional CAP line code.

Figure 12.52 CAP HDSL Functional Block Diagram



Bits to be transmitted are scrambled and passed to the bit-to-symbol mapper and Trellis encoder, which generates two multi-level symbols (in-phase and quadrature) during each symbol period. These symbols, represented by digital numbers, are passed through a Tomlinson precoder. The symbols are then passed to the CAP transmitter, which has two filters with the same passband amplitude response, but with phase responses that differ by 90 degrees. The outputs of the two filters (in-phase and quadrature) are summed, filtered, and passed to a hybrid circuit. This hybrid circuit passes the transmitted signal to the twisted-pair subscriber line while taking the received signal from the other end of the subscriber line and passing it to the receiver. Additionally, there is a balance circuit (B) that maximizes hybrid (echo path) loss.

The received signal is passed from the hybrid circuit through the receive front end. The receive front end generally has a filter to band limit the noise, as well as a programmable gain amplifier and a converter to produce the digital words from the analog received signal.

Next, a replica of the transmitted echo is subtracted from the digital signal, and the remaining signal is passed to the receiver. The receiver performs the necessary equalization as well as Viterbi decoding of the received signal. Finally, the signal is descrambled and passed on as the received bits.

The bit rate on each subscriber line is 784 kbps. The line format is Trellis-coded 64-CAP with Tomlinson precoding. A 2D (two-dimensional), eight-state Trellis code is used; however, the use of an optional 4D (four-dimensional), 16-state Trellis code is permitted. In the 2D, eight-state Trellis code, of the six bits per symbol, five are information bits, and one bit of redundancy is added by the 2D, eight-state Trellis code.

The symbol rate is 784/5 or 156.9 kbauds for the 2D, eight-state Trellis code. In the 4D, 16-state Trellis code, of the six bits per symbol, 5.5 are information bits, and 0.5 bits of redundancy are added by the 4D, 16-state Trellis code. The symbol rate is 784/5.5 or 142.54 kbauds for the 4D, 16-state Trellis code.

CAP Transmit Power Mask

Figures 12.53 and 12.54 provide templates for the CAP HDSL transmit signal spectrum. These power mask templates define the passband section and the upper and lower stop band sections of the spectrum.

Figure 12.53 CAP HDSL Transmit Signal Spectrum for the 4D, 16-State Trellis Code

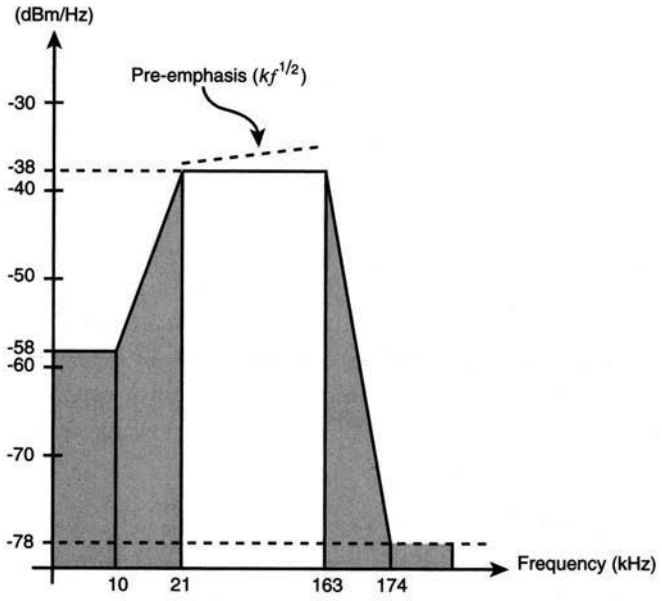


Figure 12.54 CAP HDSL Transmit Signal Spectrum for the 2D, Eight-State Trellis Code

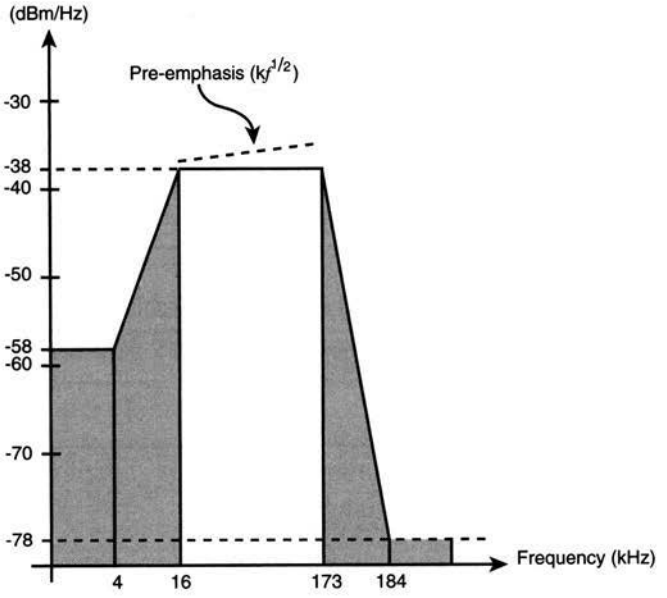


Figure 12.53 shows the transmit signal spectrum for the CAP HDSL transmitter using the 4D, 16-state Trellis code, while Figure 12.54 shows the transmit signal spectrum when using the 2D, eight-state Trellis code. The nominal shape of the transmit signal spectrum is that of the square root of a raised-cosine shape with a nominal 15-percent excess bandwidth.

The passband for any CAP implementation using the 4D, 16-state Trellis code lies in the frequency band between 20 kHz and 165 kHz, not including the excess bandwidth. The nominal –3 dB frequencies are approximately 21 kHz and 163 kHz. The spectrum is centered around a frequency of 91.96 kHz. Ripple within the passband is less than 0.5 dB. The passband for any CAP implementation using the 2D, eight-state Trellis code lies in the frequency band between 15 kHz and 175 kHz, not including the excess bandwidth. The nominal –3 dB frequencies are approximately 16 kHz and 173 kHz. The spectrum is centered around a frequency of 94.16 kHz. Ripple within the passband is less than 0.5 dB.

The signal amplitude at frequencies in the lower stop band is at least 20 dB below the nominal signal amplitude at frequencies in the passband. The lower stop band includes those frequencies below the 15-percent excess bandwidth portion shown in Figures 12.53 and 12.54. This lower stop band is not below 10 kHz. The signal amplitude at frequencies in the upper stop band is at least 40 dB below the nominal signal amplitude at frequencies in

the passband. The upper stop band includes those frequencies above the 15-percent excess bandwidth portion shown in Figures 12.53 and 12.54. For the 4D, 16-state Trellis code, this upper stop band does not exceed 174 kHz. For the 2D, eight-state Trellis code, this upper stop band does not exceed 184 kHz.

The transmit signal might incorporate some pre-emphasis in its signal spectrum to provide compromise equalization. Within the passband, the nominal spectrum might include a signal gain bound by a curve that increases at the rate shown by the dashed curve in Figures 12.53 and 12.54. The scaling factor k in the pre-emphasis is to be determined such that the overall signal power is 13.5 dBm nominal.

The average transmit power at the transmitter output is between 13.0 and 14.0 dBm into a 110 ohm termination impedance.

CAP Scambler and Descrambler

The scrambler and descrambler included in each transceiver are different for the two directions of transmission. The generating polynomials are

$$1 + x^{-18} + x^{-23}$$

for the customer premises transceiver and

$$1 + x^{-5} + x^{-23}$$

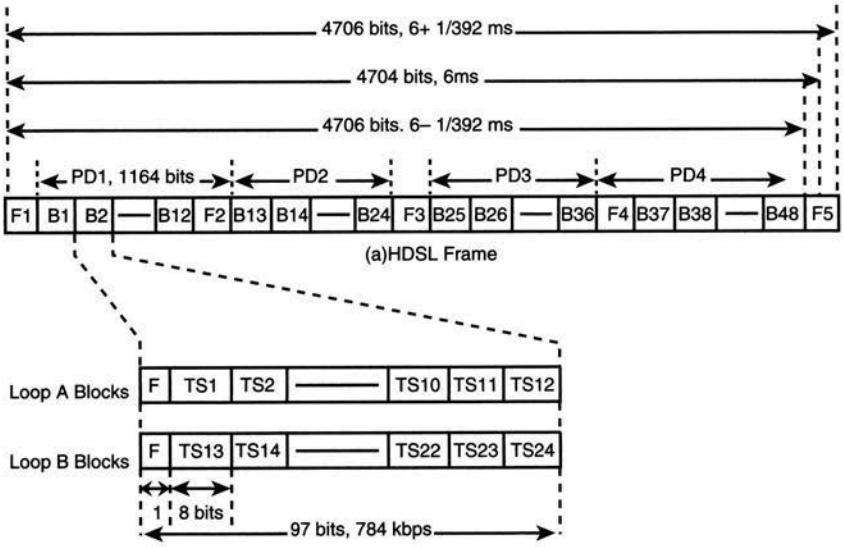
for the Central Office transceiver.

During startup, the scramblers and descramblers operate in self-synchronizing mode. During data transfer, the scramblers are locked, and the scrambled sequence is added at the transmitter and subtracted (modulo 2) at the receiver. The transfer from self-synchronizing mode to locked mode occurs with the transmit data being all 1s. Note that all bits are scrambled including the overhead bits, frame alignment word, and stuff/delete bits.

CAP Frame Structure

Figure 12.55 shows the frame format structure. The basic structure is a frame of 6 milliseconds duration that is divided into four payload data blocks (PD1–PD4) and five overhead framing blocks (F1–F5). Each 6 ms frame contains 4,704 bits, which corresponds to a nominal line bit rate of 784 kbps in each loop. Both loops have the same frame structure, but have different identifiers.

Figure 12.55 The CAP HDSL Frame Structure

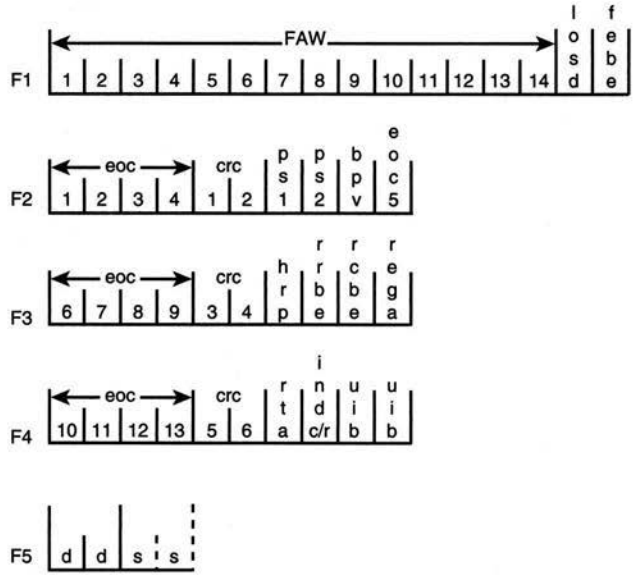


Within each payload data block are twelve 97-bit sub-blocks. Each frame contains 48 sub-blocks labeled B1–B48. Figure 12.55 also shows the sub-block data structure with the corresponding DS1 payload bits. The sub-blocks in the loop A frame contain DS0s 1–12 plus the DS1 frame (F) bit. Likewise, the sub-blocks in the loop B frame contain DS0s 13–24 plus the DS1 F bit. Note that the DS1 F bit is transmitted in both loops. Figure 12.56 shows the structure of the overhead framing blocks.

The bit definitions are described as follows:

- *Block F1.* Contains a 14-bit Frame Alignment Word (FAW) or sync word for identifying the beginning of the frame, one loss-of-line signal indicator bit (losd), and one bit indicating far-end block error (febe). Loops A and B (refer to Figure 12.55) each have a unique 14-bit FAW.
- *Block F2.* Contains four embedded operations channel bits (eoc1–eoc4), two crc-6 bits (crc1–crc2), power status bits 1 and 2 (ps 1 and ps 2), one bipolar violation bit (bpv), and one additional eoc bit (eoc5).
- *Block F3.* Contains four eoc bits (eoc6–eoc9), two crc-6 bits (crc3–cro4), one regenerator present bit (hrp), one regenerator remote block error (rrbe), one regenerator central block error (rcbe), and one regenerator alarm bit (rega).

Figure 12.56 Frame Sub-Blocks



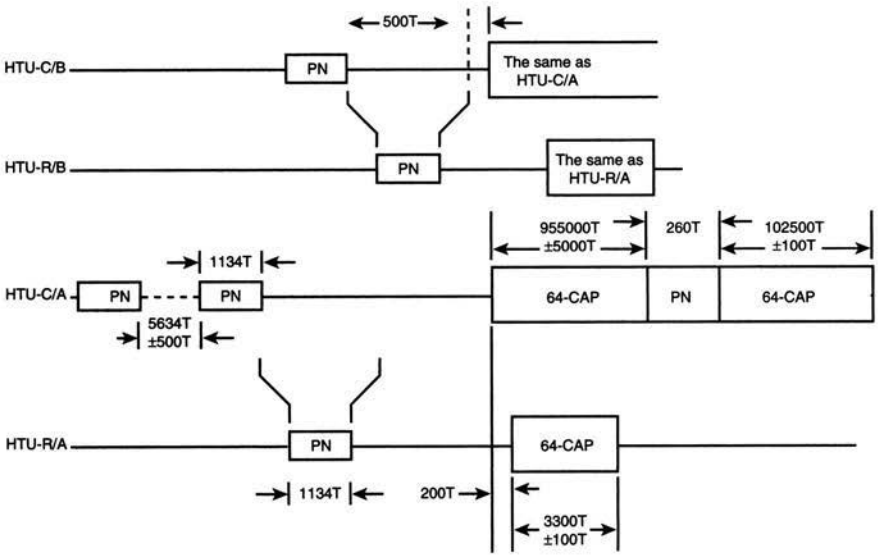
- *Block F4.* Contains the remaining four eoc bits (eoc10–wc13), the remaining two crc-6 bits (crc5–crc6), one remote terminal alarm bit (rta), one ready-to-receive bit (indelindr), and two undefined bits (uib).
- *Block F5.* Contains either zero, two, or four stuff bits.

CAP HDSL Startup

The startup procedure for CAP HDSL transceivers consists of four phases (as illustrated in Figure 12.57):

1. Alerting
2. Front-end training
3. Echo canceller and equalizer training
4. Tomlinson coefficient exchange

Figure 12.57 CAP HDSL Startup Sequences



Details of the CAP HDSL startup sequence can be found in the ANSI HDSL technical report.

Relying on extensive decision feedback equalization and echo cancellation, HDSL can provide a transmission throughput of 800 kbps at a signal spectral bandwidth of 200 kHz over CSA loops. As lately discovered during the study of Asymmetric Digital Subscriber Lines (ADSL), the frequency band above 200 kHz can be very useful if the NEXT can be avoided through Frequency Division Multiplex (FDM).

The use of FDM can achieve a higher transmission throughput potential while saving the signal processing of digital adaptive echo cancellation. The development of single-pair HDSL, also called *HDSL2* or *next generation HDSL,* has explored the benefit of minimizing the effect of NEXT in conjunction with the use of some advanced Trellis coding techniques.

End Notes

1. J. W. Lechleider, "The Feasibility and Advisability of a DSL Operating at Substantially Higher Information Rates Than That of the Basic Access DSL Should Be Studied," *Bellcore T1E1 Contribution,* T1E1.4/88-038 (1988).

2. J. W. Lechleider, "Some Areas for Further Study for HDSL," *Bellcore T1E1 Contribution,* T1E1.4/88-144 (July 25, 1988).

3. R. A. McDonald, "Suggestions for the Study of the HDSL," *Bellcore T1E1 Contribution,* T1E1.4/89-065 (1989).

4. R. A. McDonald and C. F. Valenti, "Assumptions for Bellcore HDSL Studies," *Bellcore T1E1 Contribution,* T1E1.4/89-066 (1989).

5. C. F. Valenti, "Margin Considerations for HDSL Studies," *Bellcore T1E1 Contribution,* T1E1.4/89-232 (1989).

6. P. Adams and J. Cook, "The Specification of NEXT Margin for a High Rate DSL," *BT Labs T1E1 Contribution,* T1E1.4/89-261 (1989).

7. J. W. Lechleider, "The Range of DSLs Using a Single Pair and Operating at 1.544 Mbps," *Bellcore T1E1 Contribution,* T1E1.4/89-070 (1989).

8. J. W. Lechleider, "The Performance of the 2B1Q Line Code at 800 kbps in a Carrier Serving Area," *Bellcore T1E1 Contribution,* T1E1.4/89-068 (July 25, 1989).

9. J. W. Lechleider, "The Application of Trellis Coded Modulation to a Population of Carrier Serving Area Loops," *Bellcore T1E1 Contribution,* T1E1.4/89-069 (1989).

10. V. B. Lawrence, J. J. Werner, and N. Zervos, "HRDSL Line Code: Fundamental Limits," *AT&T Bell Labs T1E1 Contribution,* T1E1.4/89-103 (1989).

11. E. Arnon, "High Rate DSL System Reach with Enhanced Feed Forward Filtering," *Northern Telecom T1E1 Contribution,* T1E1.4/89-085 (1989).

12. E. Arnon and E. Ehrlich, "Cable Loss Distribution in a Carrier Serving Area," *Northern Telecom T1E1 Contribution,* T1E1.4/89-085 (1989).

13. K. Sistanizadeh, "The Performance of an Ideal Decision Feedback Filter on HDSL," *Bellcore T1E1 Contribution,* T1E1.4/89-234 (1989).

14. G. E. Smith, "High Rate DSL Line Code: Proposal for Coded QAM," *AT&T Bell Labs T1E1 Contribution,* T1E1.4/89-151 (1989).

15. G. E. Smith, "Performance of Coded QAM in the HDSL Environment," *AT&T Bell Labs T1E1 Contribution,* T1E1.4/89-199 (1989).

16. C. F. Valenti and R. A. McDonald, "Two Pair Full-Duplex vs. Two Pair Simplex Loss and Range Consideration," *Bellcore T1E1 Contribution,* T1E1.4/89-233 (1989).

17. J. A. C. Bingham, "Duplex Systems and Signal Formats for HDSL," *Telebit T1E1 Contribution,* T1E1.4/89-242 (1989).

18. R. A. McDonald, "Technical Report: Study of the Feasibility and Advisability of Digital Subscriber Lines Operating at Rates Substantially in Excess of the Basic Access Rate," T1E1.4/90-002 (1990).

19. T. Starr and M. Bono, "Proposed HDSL Cable Models," *Ameritech T1E1 Contribution,* T1E1.4/89-243 (1989).

20. T. Starr and M. Bono, "Proposed HDSL Cable Models and Delay Limits," *Ameritech T1E1 Contribution,* T1E1.4/90-012 (1990).

21. N. Cole and N. Lynch-Aird, "Calculated NEXT-Limited Reach of a Simple 800 kbps HDSL System in the Presence of ISDN Basic Rate and T1 Interferes," *BT Labs T1E1 Contribution,* T1E1.4/89-262 (1989).

22. N. Cole and N. Lynch-Aird, "Re-calculated NEXT-Limited Reach for Simple 800 kbps HDSL Systems in the Presence of ISDN Basic Rate and T1 Interferes," *BT Labs T1E1 Contribution,* T1E1.4/90-027 (1990).

23. E. Anon, "Cable Loss Distribution in Carrier Serving Area," *Northern Telecom T1E1 Contribution,* T1E1.4/90-053 (1990).

24. G. Forney, Jr., "Coding for Band-Limited Channels," *Motorola Codex T1E1 Contribution,* T1E1.4/90-084 (1990).

25. M. V. Eyuboglu, G. Pottie, and L. Brown, "Combined Coding and Precoding for HDSL," *Motorola Codex T1E1 Contribution,* T1E1.4/90-059 (March 1990).

26. M. V. Eyuboglu, G. Pottie, and L. Brown, "Performance of Combined Coding: Precoding in the Presence of Channel Variations," *Motorola Codex T1E1 Contribution,* T1E1.4/90-085 (June 1990).

27. N. J. Lynch-Aird, "Tomlinson Precoding and DFE PAM: An Analytical Comparison for Recursive and Nonrecursive Equalizers," *BT Labs T1E1 Contribution,* T1E1.4/90-146 (1990).

28. M. Sorbara, J. J. Werner, and N. A. Zervos, "Carrierless AM/PM Tutorial," *AT&T Bell Labs T1E1 Contribution,* T1E1.4/90-154 (1990).

29. G. J. Pottie, "Comparison of Baseband vs. Passband Systems for HDSL," *Motorola Codex T1E1 Contribution,* T1E1.4/90-164 (1990).

30. K. Sistanizadeh, "Performance Evaluation Studies of QAM and PAM Schemes on a Sample of CSA Loops with an Ideal DFE," *Bellcore T1E1 Contribution,* T1E1.4/90-180 (1990).

31. E. Anon, "Impulse Noise Immunity of HDSL Systems," *Northern Telecom T1E1 Contribution,* T1E1.4/90-170 (1990).

32. K. J. Kerpez, "2B1Q and QAM Performance in the Presence of Impulse Noise," *Bellcore T1E1 Contribution,* T1E1.4/90-179 (1990).

33. D. L. Waring and K. Sistanizadeh, "Suggested Plan to Assess and Select HDSL Transmission Schemes for Further Study," *Bellcore T1E1 Contribution,* T1E1.4/90-109 (1990).

34. R. C. McConnell, "HDSL Applications with DS1 Lines Having Frequency Offsets Between Direction," *Bellcore T1E1 Contribution,* T1E1.4/90-110 (1990).

35. W. Y. Chen, "Analog and Digital Solutions for Asynchronous Clocks," *Bellcore T1E1 Contribution,* T1E1.4/91-029.

36. E. Arnon and E. Ehrlich, "A Frame Format and Overhead Channel for HDSL," *BNR T1E1 Contribution,* T1E1.4/90-114 (1990).

37. W. R. Murphy, "Alternative HDSL Framing Format and Overhead Channel," *Tellabs T1E1 Contribution,* T1E1.4/90-152 (1990).

38. H. Nicholas and H. Flagg, "HDSL Frame Format," *PairGain Technologies T1E1 Contribution,* T1E1.4/90-167 (1990).

39. W Y. Chen, "A Double-Barker Code Variable Frame Format," *Bellcore T1E1 Contribution,* T1E1.4/91-109 (August 10, 1991).

40. E. Arnon and E. Ehrlich, "Measurement Results of an HDSL Transmission System," *BNR T1E1 Contribution,* T1E1.4/90-15X (1990).

41. R. Cupo and M. Sorbara, "Margin Performance of a Carrierless AM/PM HDSL System," *AT&T Bell Labs T1E1 Contribution,* T1E1.4/90-153 (1990).

42. K. T. Foster, "Crosstalk Margin Measurements for an Example 768 kbps HDSL Transmission System," *BT Labs T1E1 Contribution,* T1E1.4/90-165 (1990).

43. B. A. Blake, "Suggested Testing Plan for HDSL Prototype Transceivers," *Bellcore T1E1 Contribution,* T1E1.4/90-182 (1990).

44. B. A. Blake, "Description of a Loop Emulator Test Bed at Bellcore for Constructing Loops Out of Actual Cable in the Lab," *Bellcore T1E1 Contribution,* T1E1.4/90-183 (1990).

45. K. Sistanezadeh, "A Tentative CSA Loop Population for Preliminary Studies of HDSL Transmission Schemes," *Bellcore T1E1 Contribution,* T1E1.4/90-184 (1990).

46. D. A. Fisher and D. R. Donovan, "Advantages of 448 kbps (6B+D) Modularity and the 2B1P Line Code," *STC Technology T1E1 Contribution,* T1E1.4/90-120 (1990).

47. D. A. Fisher and D. R. Donovan, "A Proposal for 448 kbps Framing Structure," *STC Technology T1E1 Contribution,* T1E1.4/90-121 (1990).

48. J. S. Chow, J. C. Tu, and J. M. Cioffi, "A Discrete Multitone Transceiver System for HDSL Application," *Stanford University T1E1 Contribution,* T1E1.4/90-209 (December 1990).

49. J. C. Tu, J. S. Chow, and J. M. Cioffi, "A Discrete Multitone Transceiver System for HDSL Application," *Stanford University T1E1 Contribution,* T1E1.4/90-210 (December 1990).

50. K. Sistanezadeh, "Performance Variation of QAM and PAM with Respect to Source and Load Terminations on HDSL," *Bellcore T1E1 Contribution,* T1E1.4/90-213 (December 1990).

51. H. Samueli, "Design and Implementation of a 64-Tap CMOS Echo Canceller/Decision-Feedback Equalizer for 2B1Q HDSL Transceivers," *UCLA T1E1 Contribution,* T1E1.4/90-218 (December 1990).

52. "Finite-Length Decision Feedback Equalizer for HDSL," *Motorola Codex T1E1 Contribution,* T1E1.4/90-225 (December 1990).

53. H. Y. Chang and M. Sorbara, "Effects of Impulse Noise on Baseband and Passband Transmission Schemes for HDSL," *AT&T Bell Labs T1E1 Contribution,* T1E1.4/90-224 (December 1990).

54. K. Sistanezadeh and B. Blake, "A Non-CSA Loop for HDSL Performance Evaluation Studies," *Bellcore T1E1 Contribution,* T1E1.4/90-215 (December 1990).

55. M. D. Martino and W. R. Murphy, "A Format for the HDSL Framing and Overhead Channel," *Tellabs T1E1 Contribution,* T1E1.4/90-220 (December 1990).

56. M. Sorbara, "Frame Format for a Passband HDSL," *AT&T Bell Labs T1E1 Contribution,* T1E1.4/90-223 (December 1990).

57. "HDSL Framing Overhead Ad Hoc Working Group Report," T1E1.4/90-238 (December 1990).

58. T. Starr, "Baseband Analog Voice Transmission for HDSL," *Ameritech T1E1.4 Contribution,* T1E1.4/91-013 (February 1991).

59. R. C. McConnel, "Example HDSL Framing and Overhead Structure," *Bellcore T1E1 Contribution,* T1E1.4/91-021 (February 1991).

60. H. Flagg, "Frame Structure for a High-Rate Digital Subscriber Line," *PairGain T1E1 Contribution,* T1E1.4/91-036 (February 1991).

61. D. Amrany and S. Gadot, "General Adaptive FIR Filter: a VLSI Chip for HDSL and ADSL," *AT&T Network Systems T1E1 Contribution,* T1E1.4/91-034 (February 1991).

62. R. Brackert and B. Itri, "Aerial Cable Performance on a 9 kft Loop: A Study of Environment Temperature Changes on an HDSL Transceiver," *PairGain Technologies T1E1 Contribution,* T1E1.4/91-061 (February 1991).

63. B. A. Blake, "NEXT Performance Margin Test Results for a 3B1O HDSL Transceiver Demonstration System from British Telecom Research Laboratories," *Bellcore T1E1 Contribution,* T1E1.4/91-023 (February 1991).

64. B. A. Blake, "NEXT Performance Margin Test Results for a 2B1Q HDSL Transceiver Prototype from Bell-Northern Research," *Bellcore T1E1 Contribution,* T1E1.4/91-024 (February 1991).

65. B. A. Blake, "NEXT Performance Margin Test Results for a Carrierless AM/PM HDSL Transceiver Prototype from AT&T," *Bellcore T1E1 Contribution,* T1E1.4/91-025 (February 1991).

66. B. A. Blake, "NEXT Performance Margin Test Results for a 2B1Q HDSL Transceiver Prototype from Tellabs," *Bellcore T1E1 Contribution*, T1E1.4/91-026 (February 1991).

67. B. A. Blake, "NEXT Performance Margin Test Results for a 2B1Q HDSL Transceiver Prototype from PairGain Technologies," *Bellcore T1E1 Contribution,* T1E1.4/91-027 (February 1991).

68. R. A. McConnel, "Proposal to Change the Name of HDSL Technical Report," *Bellcore T1E1 Contribution,* T1E1.4/91-033 (February 1991).

69. Ad Hoc Group on HDSL Framing and Overhead, "Ad Hoc Group Meeting on HDSL Framing and Overhead and Outline for Text on Alternative Transmission Techniques," T1E1.4/91-046 (February 1991).

70. W.Y.Chen, J.L.Dixon, and D.L.Waring, "HDSL Echo Cancellation," *IEEE JSAC,* vol. 9, no. 6 (August 1991): 848–860.

71. K. H. Mueller and M. Muller, "Timing Recovery in Digital Synchronous Data Receivers," *IEEE Trans. On Comm.*, vol. COM-24, no. 5 (May 1976): 516–531.

72. H. Samueli, B. Daneshrad, R. B. Joshi, B. C. Wong, and H. T. Nicholas, III, "A 64-Tap CMOS Echo Canceller/Decision Feedback Equalizer for 2B1Q HDSL Transceivers," *IEEE JSAC,* vol. 9, no. 6 (August 1991): 839–847.

73. K. Hwang, *Computer Arithmetic, Principles, Architecture, and Design* (New York: Wiley, 1979).

74. N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design* (Reading, Mass.: Addison-Wesley, 1985).

75. K. Sistanizadeh, "A Tentative CSA Loop Population for Preliminary Studies of HDSL Transmission Schemes," *Bellcore T1E1 Contribution,* T1E1.4/90-184 (1990).

76. ANSI T1E1.4 Working Group on Digital Subscriber Lines, "A Technical Report on High-bit-rate Digital Subscriber Lines (HDSL)," T1E1.4/96-006 (April 22, 1996).

77. Test sets from TAS, **http://www.taskit.com/**.

78. Test sets from Consultronics, **http://www.consultronics.on.ca/**.

79. W. Y. Chen, "Choices of HDSL Load Resistor Values," *Bellcore T1E1 Contribution,* T1E1.4/91-078 (May 13, 1991).

80. R. C. McDonnel, "HDSL Application with DS1 Lines Having Frequency Offsets Between Directions," *Bellcore T1E1 Contribution,* T1E1.4/90-110 (June 18, 1990).

81. R. C. McDonnel, "Private Digital Network Frequency Synchronization with Local Exchange Carriers," *Bellcore T1E1 Contribution,* T1E1.4/90-217 (December 10, 1990.)

82. S. A. Cox, "Clock Sensitivity Reduction in Echo Cancellers," *Electron. Lett.*, vol. 21 (July 4, 1985).

83. D. G. Messerschmitt, "Asynchronous and Timing Jitter Insensitive Data Echo Cancellation," *IEEE Trans. Com,-nun.*, vol. COM-34, no. 12 (December 1986): 1209–1217.

84. J-J. Werner, "An Echo-Cancellation-Based 4800 Bit/s Full-Duplex DDD Modem," *IEEE Trans. JSAC,* vol. SAC-2, no. 5 (September 1984): 722–730.

85. J. C. Candy and 0. J. Benjamin, "A Circuit That Changes the Word Rate of Pulse Code Modulated Signals," *Bell System Tech. J.*, vol. 62, no. 4 (April 1983): 1161–1168.

86. C. W Farrow, "A Continuously Variable Digital Delay Element," *Proc. IEEE ISCAS '88* (1988): 2641–2645.

87. W. Y. Chen, "Analog and Digital Solutions for Asynchronous Clocks," *Bellcore T1E1 Contribution,* T1E1.4/91-029.

88. M. D. Martino, W. R. Murphy, and H. Flagg, "Proposed Format for the HDSL Framing and Overhead Channel," *Tellabs and PairGain Contribution,* T1E1.4/91-085 (May 13, 1991).

89. E. Arnon, "Frame Format for HDSL," *Northern Telecom Contribution,* T1E1.4/91-082 (May 13, 1991).

90. Conversation with Michael Martino of Tellabs.

91. W Y. Chen, "A Double-Barker Code Variable Frame Format," *Bellcore T1E1 Contribution,* T1E1.4/91-109 (August 10, 1991).

92. W Y. Chen, "The Performance of the Double-Barker Code Under Impulse Noise," *Bellcore T1E1 Contribution,* T1E1.4/91-110 (August 10, 1991).

93. *Digital Hierarchy-Formats Specifications*, ANSI T1.107-1988 (1988).

94. *Digital Hierarchy-Electrical Interfaces*, ANSI T1.102-1989 (1989).

95. *Carrier-to-Customer Installation: DS1 Metallic Interface*, ANSI T1.403-1989 (1989).

96. *ISDN Primary Rate: Customer Installation Metallic Interfaces Layer 1 Specification*, ANSI T1.408-1990 (1990).

97. T1E1.4 Working Group on Digital Subscriber Lines, R. C. McConnell and B. H. Soloway, Ed., "High-bit-rate Digital Subscriber Lines," Working Draft, *Technical Report Issue 2*, T1E1.4/96-006 (April 22, 1996).

98. B. A. Blake and W. Y. Chen, "Suggested Testing Plan for HDSL Prototype Transceivers," *Bellcore T1E1 Contribution,* T1E1.4/90-182 (September 26, 1990).

99. B. A. Blake and W. Y. Chen, "Amendment to Suggested Testing Plan for HDSL Prototype Transceivers," *Bellcore T1E1 Contribution,* T1E1.4/90-216 (December 10–14, 1990).

CHAPTER 13

ADSL (Asymmetrical Digital Subscriber Lines)

The desire for the further advancement of telephone subscriber loop-based DSL technology was very strong after the initial definition of the dual-duplex HDSL as a repeaterless T1 technology. The performance improvement of HDSL was partially due to its broader signaling bandwidth compared with that of DSL. Therefore, the next DSL technology after HDSL should utilize more bandwidth than that used by HDSL.

On the other hand, the NEXT noise could have made the telephone subscriber loop channel useless beyond the frequency band of HDSL even at CSA range. The only possibility to further expand the usable bandwidth is to avoid the NEXT noise by adopting one directional transmission among all telephone subscriber loops.

Consumer broadband services that could benefit from DSL technologies had been considered video oriented. Video telephony was a candidate, but its market potential was questionable based on previous attempts. The video-on-demand service had been considered as a telephone company's response to video rental and cable television pay-per-view markets. The video-on-demand service can be provided through a T1 rate (1.544 Mbps) DSL system in conjunction with a digital video compression technique, such as MPEG1 from a CO to a telephone subscriber. In the reverse direction, only a lower throughput channel is necessary to send ordering information as well as real-time control commands similar to that of a VCR.

Asymmetrical Digital Subscriber Lines (ADSL) were originally developed for such a video-on-demand service with a downstream (from a CO to a subscriber) transmission throughput of 1.544 Mbps and an upstream (from a subscriber to a CO) transmission throughput of between 16 and 64 kbps. With ADSL, one compressed video channel can be provided to all consumers connected with nonloaded telephone subscriber loops. At CSA range, up to four

compressed video channels can be provided. Meanwhile, Trellis and forward error correction coding techniques had been introduced in conjunction with the Discrete MultiTone (DMT) line code to improve the transmission performance under the crosstalk and impulse noise environment.

Most recently, ADSL has been considered as the next-generation transmission technology for high-speed data access applications. The ever increasing population of home PCs and the widespread use of the Internet create a demand for a higher access speed than what a voice band modem can provide. Because of the random nature of data communication and corresponding transmission protocol, some ADSL transmission overhead designed for multiple video channels can be simplified. The adoption of an ATM-oriented transmission interface is another flexible approach to make ADSL technology suitable for different applications.

This chapter covers development activities, line codes, transmission performances, and interface standards of Asymmetrical Digital Subscriber Lines, including the following:

- *Development of ADSL technology.* Technical discussions leading to the development of ADSL are reviewed. Early performance measurements on prototype ADSL systems and in conjunction with a POTS service are presented.
- *ADSL transceiver structures and line codes.* For the ADSL standard defined by T1.413, the general ADSL transceiver architecture and issues, such as TEQ, FEQ, and sub-carrier SNR, are discussed. Transmission parameters for both downstream and upstream directions are summarized. ADSL specific techniques including Trellis coding, Reed-Solomon coding, interleaving, and asymmetric echo cancellation are reviewed.

  Implementation strategy, requirement, and complexity are addressed with respect to digital interface, digital signal processing, and analog interface functional blocks for both CO and subscriber side transceivers. Transceiver structures, transmission parameters, specifics of transmit filter and equalizer coefficients, and possible implementation approaches are also discussed for CAP line code ADSL.
- *ADSL performance objectives.* An additional six ADSL mid-CSA test loop configurations and corresponding transfer functions as well as three noise models with combined NEXT, FEXT, and AWG noises are presented. DMT ADSL transmission performances are simulated for three different ranges of test loops with associated noise conditions. Transmission performances of CAP ADSL are also estimated.
- *ADSL network and premises interfaces.* ADSL network interface at a CO or a RT for providing video on demand services is discussed. Associated ADSL multi-channel specifications are reviewed. ADSL network interface for providing data access services is also

discussed. The function and the design of POTS splitter is addressed. Possible approaches of POTS splitter installation in conjunction with existing telephone in-house wiring are presented.

- *ADSL standards.* Key features of ADSL defined by ANSI T1.413, such as frame/superframe structure, channelization, bit-to-constellation mapping, and initialization procedures, are discussed.

13.1 Development of ADSL Technology

The development of HDSL is relatively straightforward. Based on experience gained from DSL development, people involved felt that the same signal processing and *Very-Large-Scale Integrated circuit* (VLSI) technologies could be enhanced for a higher transmission throughput. The next logical step of higher throughput is the T1 rate: 24 B channels at a rate of 1.544 Mbps. Computer performance simulation studies indicated that two pairs of cables were required.

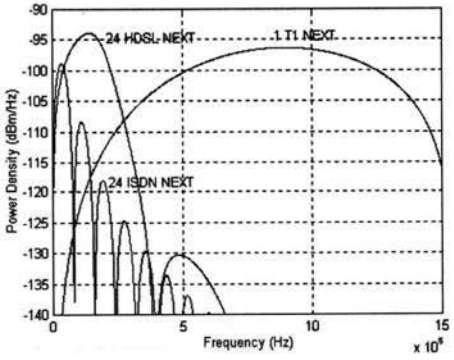
After the demonstration of the prototype VLSI chip, the echo cancellation-based 2B1Q line code was selected for HDSL. HDSL then found an application as the dual-duplex repeaterless T1 technology. To avoid the lengthy standardization process experience of DSL, it was decided that a technical report by the T1E1.4 working group would be sufficient to guide HDSL vendors for the compliance of interoperability.

During these performance studies, two other line codes—Carrierless AM/PM (CAP) and Discrete MultiTone (DMT)—were also proposed as alternatives. Due to the lack of understanding of these alternative line codes by the majority of standards activity participants, these line codes were not favored for HDSL. Due to persistent efforts of their proponents as well as the availability of corresponding tutorial material, however, the T1E1.4 working group recognized the potential of using CAP and DMT line codes for the next generation of higher throughput DSL systems. The early market acceptance of HDSL by the business community further enhanced the desire for the development of a higher throughput digital subscriber line system for the mass consumer market.

HDSL is somewhat restrictive for residential services.[1] The need for two pairs and the CSA range limitation may prove prohibitive in certain areas where a significant portion of customers are beyond 12 kft or where the number of pairs-per-living unit are less than two. Thus, the genesis for the next generation of high throughput DSL systems was to design a new transport access capability to provide economic wideband services directly to the residential customer. Objectives are to cover most of the nonloaded loop plant out to 18 kft, as is the case for Basic Rate Access. In addition, it is highly desirable to reduce the number of pairs required because pairs-per-living unit ratios can be as low as 1.3 in some areas.

Although intended for the residential market, the higher throughput DSL system under consideration then needed to co-exist with early developed DSL and HDSL systems. To be conservative, the co-existance with T1 lines using the traditional transceiver technology also needed to be considered. In other words, the system under consideration then needed to operate with DSL, HDSL, and T1 NEXT. The level of crosstalk noise depend on the number of disturbers. A point of contention is that in a residential environment, the number of deployed HDSLs might be smaller. According to Figure 13.1, the new system under development could have a transmit spectrum above that of HDSL to realize the full potential of the available bandwidth of the copper loop plant.

Figure 13.1 Noise Environment with DSL, HDSL, and T1



Observing that many services needed greater transmission rates from the central office to the customer than in the other direction, Dr. Joseph Lechleider proposed the idea of asymmetrical DSLs (ADSL).[2] Adding asymmetry to a DSL system opened a new dimension to be considered with system configuration. Lechleider mainly considered 18 kft range, 2B+D full duplex, and 1.544 Mbps downstream; CSA range, 2B+D full duplex, and 3.088 Mbps downstream; and 5 kft range, 1.544 Mbps upstream, and 6.176 Mbps downstream single pair as well as a few other dual-pair ADSL options.

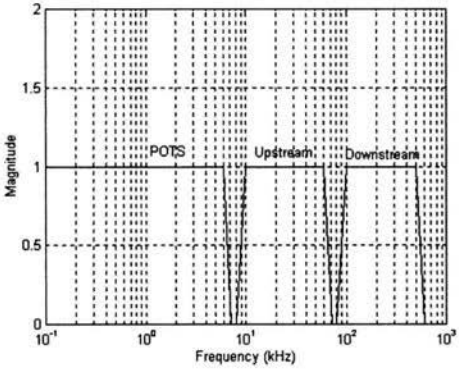
For the first two options, Lechleider assumed a unilateral downstream channel occupying a frequency band above 75 kHz in addition to DSL. For the third option, Lechleider assumed a unilateral downstream channel occupying a frequency band above 425 kHz in addition to a double baud rate HDSL. The higher transmission throughput of unilateral channels are possible because they are limited only by effect of Far End Crosstalk (FEXT).

This concept of ADSL fits the concept of a video-on-demand service provided through the existing copper loop plant very well. The CCITT H.261 recommendation and the ISO Moving Picture Experts Group (MPEG) recommendation both provide full-motion video at rates of around 1.3 Mbps. The video, an audio channel, and associated overhead can be transmitted within a 1.5 Mbps signal. Users could signal the network, scan archives of programming, and receive video-on-demand. For this service, only a low transmission throughput upstream control channel is necessary to send back control signals for interactive commands, such as Pause and Play.

The concept of ADSL also fits applications, such as network computing where software and database records could be stored on network servers and retrieved at a speed equivalent to the CD-ROM and tele-education where a specialist can be shared with a large student population through downstream channels and individual feedback can be provided through upstream channels. It was realized that for residential applications, a POTS service—with which the traditional telephone service can be provided regardless of whether the ADSL was on or off—was more practical than the ISDN service.

After sharing ideas among Bellcore, regional Bell operating companies, and other industry and academic leading ADSL proponents, ADSL was proposed as having the traditional duplex POTS channels, occupying the frequency band between 300 Hz and 4 kHz, an upstream control channel of 16–64 kbps, occupying the frequency band between 10 kHz to 50 kHz, and a downstream channel of 1.544 Mbps, occupying the frequency band of 100 kHz–500 kHz. Figure 13.2 shows the initial frequency spectra for ADSL technology.

Figure 13.2 Initial ADSL Frequency Spectra



13.1.1 Convergence of Initial Ideas on ADSL

The discussion about ADSL at T1E1.4 working group was not started until after the proposal of a new study project about ADSL by U.S. West during the May 1991 meeting.[3] During the August 1991 T1E1.4 meeting, the performance of a QAM ADSL was studied in conjunction with DSL NEXT,[4,5] 18 test loops were proposed for ADSL,[6] the effect of impulse noise was examined,[7] and the performance of a multi-carrier system, DMT, was compared with that of a single-carrier system, QAM.[8]

During the November 1991 T1E1.4 meeting, performance of DMT was further discussed,[9,10,11] the nature of burst error was discussed,[12] the forward error correction code was proposed for ADSL,[13,14] the feasibility of an ADSL control channel was discussed,[15] and the use of Trellis code for ADSL was discussed.[16]

During the February 1992 T1E1.4 meeting, the similarity between CAP and QAM was explored,[17] and the performances of DMT and QAM were further studied.[18,19,20,21] During the May 1992 T1E1.4 meeting, ADSL system issues—ranging from interfaces to frame format—were discussed,[22,23,24,25,26,27,28] a QAM ADSL including a control channel was discussed for spectrum compatibility with other DSLs,[29,30] the ADSL Radio Frequency Interference (RFI) was discussed,[31] the POTS-generated impulse noise was discussed,[32] a performance test plan for ADSL "Olympics," was presented,[33] and the A/D precision issue was discussed.[34] The possibility of having an ADSL standard project was also discussed.[35,36]

During the August 1992 T1E1.4 meeting, the possibility of a baseband ADSL was discussed,[37] the idea of asymmetrical echo cancellation was discussed,[38] the idea of impulse noise cancellation was discussed,[39] the design issues of a CAP ADSL were discussed,[40,41] milestones of the ADSL standard were proposed,[42] possible ADSL interfaces were discussed,[43] the ADSL "Olympic" test plan was further discussed in more detail,[44,45,46] ADSL spectral compatibility was further discussed in more detail,[47,48] the possibility of an ADSL with embedded ISDN was discussed,[49] and ADSL coding and interleaving issues were further discussed.[50, 51,52,53]

During the December 1992 T1E1.4 meeting, test plans were further discussed,[54,55] Trellis coding and forward error correction coding for ADSL were discussed,[56,57,58,59] the issue of impulse noise was discussed, [60,61,62]the idea of an analytical testbed was proposed,[63] the design of a DMT transceiver was discussed,[64] the possibility of providing multiple 1.544 Mbps rates was discussed,[65,66,67] the DMT echo canceller structure was discussed,[68] the spectrum compatibility between ADSL and other DSLs was examined,[69,70] ADSL frame structure and maintenance issues were further discussed,[71,72] and possibilities of including POTS and ISDN were discussed.[73]

During the March 1993 T1E1.4 meeting, the issue of selecting forward error correction code was further discussed,[74,75,76] implementation details of the DMT ADSL were discussed,[77,78,79,80] inclusion and alternatives of providing 6 Mbps were discussed,[81,82] the effect of ADSL spectrum on T1 performance was discussed,[83] and performance measurements of ADSL prototypes were reported by Northern Telecom, Inc.[84] and Bellcore.[85,86,87,88] NTI measurements showed positive performance margins for self-FEXT (see Figure 13.3), 49-disturber ISDN NEXT (see Figure 13.4), and 24-disturber HDSL NEXT (see Figure 13.5). No error was observed when a POTS operation was performed on the same test loop.

Figure 13.3 NTI Self-FEXT Margin

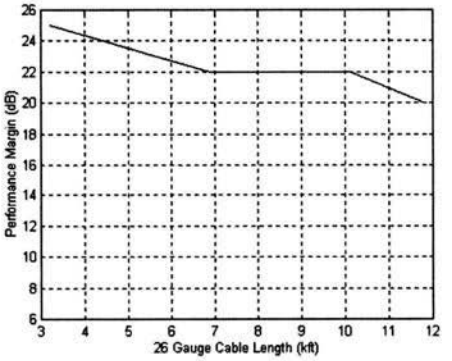


Figure 13.4 NTI 49 ISDN NEXT Margin

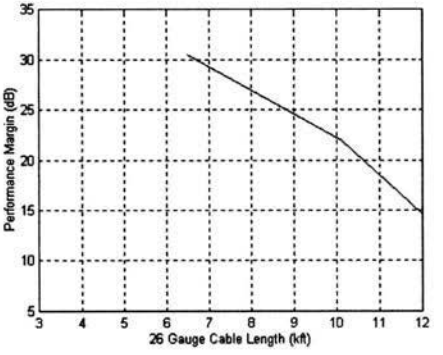
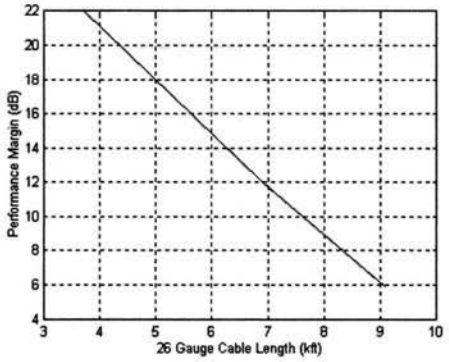


Figure 13.5 NTI 24 HDSL NEXT Margin



Bellcore ADSL "Olympic" test measurements, summarized in Table 13.1, showed Amati's ADSL prototype in the first place, Reliance/Bellcore's ADSL prototype in the second, and AT&T Paradyne's ADSL prototype in the third. The operation of the POTS sharing the same test loop also was verified for Reliance/Bellcore and Amati prototypes. Little disturbance was observed. Amati's ADSL prototype had forward error correction code. Based on these test measurements, DMT line code was selected as the ADSL standard also based on the fact that Amati's prototype could operate at 6 Mbps in addition to 1.5 Mbps.

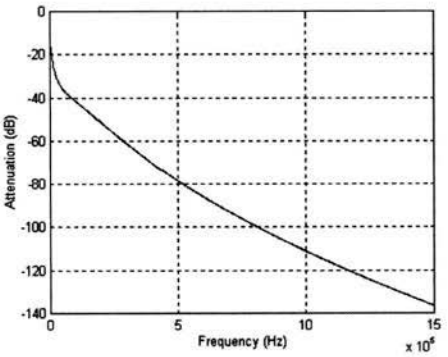
Table 13.1 Bellcore ADSL "Olympic" Test Results

Loop Number	Distur-bers	Reliance/ Bellcore	Amati 21 dBm	Amati 16 dBm	AT&T
ADSL #4	24-DSL	–6	8	4	Not work
ADSL #6	24-DSL	<–8	7	3	Not work
ADSL #9	24-DSL	7	>16	>16	Not work
ADSL #17	24-DSL	1	15	10	Not work
CSA #2	24-HDSL	>11	>11	>11	<–8
CSA #6	24-HDSL	5	>11	>11	–8
CSA #7	24-HDSL	>11	>11	>11	–1
CSA #2	1-T1	7	>10	>10	<0
CSA #6	1-T1	2	>10	8	<0
CSA #7	1-T1	8	>10	>10	–2

13.1.2 *Concerns for POTS Interference*

The feasibility of ADSL with a POTS service was established by the aforementioned prototype efforts and verified by laboratory measurements. The most obvious transmission challenge associated with the ADSL was the high level of signal loss at the extreme range of the nonloaded loop plant. Figure 13.6 shows the attenuation of an 18 kft, 24-gauge loop.

Figure 13.6 Attenuation of an 18 kft, 24-Gauge Loop



To fully realize the potential of a twisted-pair loop, the electronics front-end noise should be a few dBs below that of the twisted-pair loop plant. For an assumed twisted-pair loop plant noise power density level of −140 dBm/Hz, the RMS noise voltage for a bandwidth of 500 kHz is about 700 μV as calculated by the following expressions:

$$P = 10^{\frac{-140}{10}} \times 5 \times 10^{5} = 5 \times 10^{-9} \quad Watts$$

$$0 \qquad k \qquad \mu V$$

P is the total noise power in a bandwidth of 500 kHz, and V_{rm} is the corresponding voltage level. Within a single ADSL system itself, there is the potential for interference between the three derived channels:

- The POTS channel
- The high throughput downstream channel
- The low throughput upstream channel

The POTS channel is separated by a POTS splitter, which consists of a lowpass filter and a highpass filter. The POTS splitter design and implementation are very challenging. Ringing voltage applied at the Central Office is 86 Vrms superimposed on −48 Vdc with a two

seconds on/four seconds off pattern. When ringing is interrupted due to a telephone handset going off-hook (referred to as *ring trip*), a considerable transient can be induced in the line.

Other lower level transients are produced by dial pulsing and on-off hooks. Upstream and downstream channels can be separated by either frequency division using highpass and lowpass filters, echo cancellation, or a combination of frequency division and echo cancellation. Due to the combination of heavy channel attenuation and light echo path attenuation, a filter stop band attenuation or echo cancellation level of more than 60 dB is expected.

13.1.3 The Convergence of ADSL Standards

The first issue of ADSL standard, T1.413, was released during 1995 by the T1E1 committee. The concept of ADSL was also heavily promoted by professional technical conferences and the formation and activities of the ADSL Forum. The ADSL technology has been in technical and market trials since early 1994. The purpose of the technical trial was to verify the performance of ADSL technology in the real field. The purpose of the market trial was to get user feedback about the technology and its potential applications. Technical trials have established the viability of the ADSL technology. Market trials have likewise indicated strong user interest. Most recently, ADSL has been considered as an ideal vehicle for Internet access and telecommuniting applications.

13.2 Transceiver Structures and Line Codes

The DMT line code is a multi-carrier system. The spectrum of a DMT signal consists of many narrow bandwidth QAM-like carriers with different and equally distributed carrier frequencies. Information is attached in the transmitter and recovered in the receiver from these carriers through discrete versions of Fourier Transforms. The CAP line code, similar to QAM, is a single carrier system. Information is attached to and recovered from the single carrier through transmit filters and receiver channel equalizers. One similarity between DMT and CAP line codes is that there is no carrier modulation and demodulation required.

13.2.1 DMT Line Code

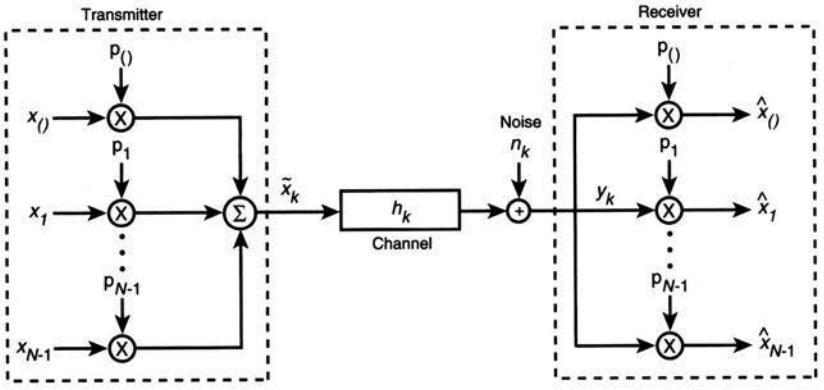
We can envision the DMT ADSL as a group of 256 voice band modems with their carrier frequencies ranging from a few kHz to near 1 MHz and with no guardband in between adjacent carriers. The need of guardband is eliminated due to the *orthogonolity* between all carriers. Similar to the voice band situation, each modem in this group can fall back to lower data rates by changing its constellation size. The capability of variable constellation size for each subcarrier is the DMT approach of dealing with a random channel with different Signal to Noise Ratios (SNRs) at different frequencies.

On the other hand, because frequency responses of all subchannels of this modem group are relatively flat, equalizers of the group of voice band modems are all omitted. The use of Trellis coding can still be applied. Because all the modems in this group can be synchronized, the transmit and receive filter functions can be combined and performed by an IFFT (Inverse Fast Fourier Transform) for all transmit filters, and an FFT (Fast Fourier Transform) for all receive filters.

On the line after the ADSL transmitter, we can imagine 256 different but synchronized signals each of which has its distinguishable carrier frequency and constellation size. The number of 256 subchannels corresponding to an IFFT/FFT size of 512 was chosen to maximize ADSL transceiver hardware efficiency while maintaining a tolerable IFFT/FFT processing delay time. A 512 real-to-complex FFT and a 512 complex-to-real IFFT were actually used.

Figure 13.7[89] shows a simplified DMT system. The major function of a DMT transmitter is the N (for which N=512) dimensional IFFT. The major function of a DMT receiver is the N-dimensional FFT.

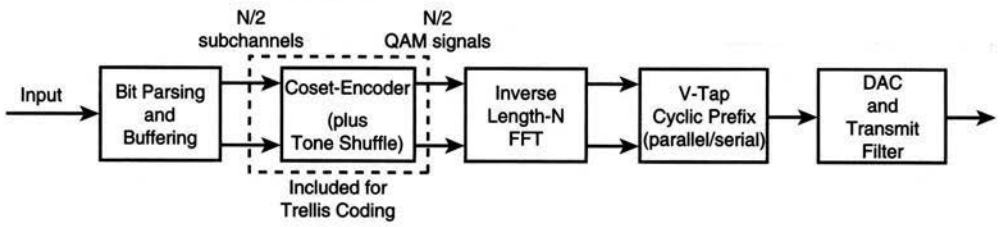
Figure 13.7 Simplified Illustration of DMT Transceiver and Channel



DMT ADSL Transmitters

The general transmitter illustrated in Figure 13.8 consists of bit parsing, buffering, coset-encoder, IFFT, cyclic prefix, parallel to serial conversion, Digital to Analog Conversion, (DAC), transmitter filter, and line interface functions.

Figure 13.8 Block Diagram of a DMT Transmitter



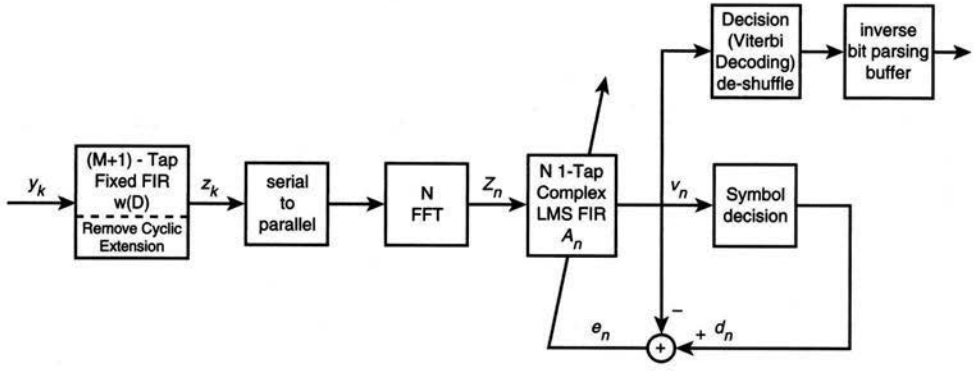
The serial bit stream is first converted into parallel data in the transmitter. The N-dimensional parallel data is then mapped into $N/2$ multibit subchannels according to a bit allocation algorithm. The coset-encoder implements the bit allocation algorithm. To achieve an efficient DFT implementation we need $N=2^i$, where i is a positive integer.

Each multibit subchannel can use a QAM signal constellation. The parallel time domain data after the IDFT is converted back to serial. A guard period stuffed with cyclic prefixes is introduced before the DAC function. The cyclic prefix is defined as the partial duplication and addition of the last v points in front of the N original data points, resulting in a total number of $v+N$ cyclic prefixed data points.

DMT ADSL Receivers

Figure 13.9 shows the general structure of a DMT receiver. The general receiver consists of a line interface, Analog to Digital Conversion (ADC), Time-Domain Equalizer (TEQ), serial to parallel conversion, FFT, Frequency-Domain Equalizer (FEQ), symbol decision, bit decision, and inverse bit parsing functions.

Figure 13.9 Block Diagram of a DMT Receiver



After ADC, the received signal is first filtered by the TEQ (the (M+1) tap fixed FIR filter, as illustrated in Figure 13.9. The samples corresponding to the guard period cyclic prefix are simply ignored. The rest of the samples are converted into parallel data for input to the DFT process. The FEQ (the N 1-tap complex FIR filter in Figure 13.9) is implemented by inverse target channel scaling. The target channel is the combination of channel and TEQ. FEQ is necessary to correct both magnitude and phase distortion of the equalized channel when using a single decision device for all subchannels.

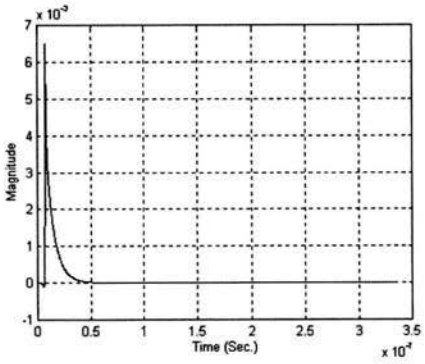
Note

It is worthwhile to mention that the purpose of the TEQ is to contain the amount of channel distortion in the time domain. The use of guard periods is to avoid the residual inter-baud-symbol interference. Residual channel distortion, in the form of different magnitude attenuation and phase delay for each subcarrier, however, still exists for each baud symbol.

Combating Channel Distortion

In theory, the channel distortion can be avoided by a DMT system through the fine division of the available bandwidth into many subcarriers such that each subchannel is almost distortion free. The finer division in frequency domain is equivalent to further extension of the signaling baud period. The channel distortion can be considered negligible if the channel impulse response tail caused by channel distortion is negligible compared with the signaling baud period. Figure 13.10 shows the channel impulse response of the 9 kft, 26-gauge loop.

Figure 13.10 Channel Impulse Response of a 9 kft, 26-Gauge Loop



The tail length is about 2.77 microseconds at 1 percent of the peak amplitude. The signaling baud length can be chosen as 277 microseconds for a factor of 100 of the tail length. The baud rate is then 3.61 kHz. On the other hand, the overall delay time, which is about three times the baud length, should also be constrained according to the system requirement.

The channel distortion could cause inter-subchannel interference. The inter-subchannel interference is caused by amplitude attenuation and phase delay differences between subchannels. The total inter-subchannel interference for a particular subchannel can be calculated as follows:

$$10\log_{10}\sum_{\substack{k=1\\k\neq i}}^{N}\int_{0}^{T}\cos\left(2\pi i\frac{t}{T}-\theta_i\right)\cdot\frac{1-\Delta A_k}{1-\Delta A_i}\left(\cos\left(2\pi k\frac{t}{T}-\theta_k\right)+\sin\left(2\pi k\frac{t}{T}-\theta_k\right)\right)dt$$

The values are as follows:

ΔA_i and ΔA_k are the subchannel and interfering subchannel amplitude attenuations, respectively.

θ_i and θ_k are the subchannel and interfering subchannel phase delays, respectively.

The inter-subchannel interference can be minimized by selecting a lower baud rate or by using a channel equalizer.

Although the effect of the channel impulse response tail is minimized through the selection of a slow baud rate, the tail of one baud will extend to the next adjacent baud. This inter-baud interference is the DMT inter-symbol interference. The effect of DMT inter-symbol interference can be minimized by using a guard period or a cyclic prefix.[90] The length of a cyclic prefix is chosen such that the tail portion of one baud will die out before the next adjacent baud starts. The cyclic prefix repeats the last v time domain serial data points before the first serial data point, in which v is the length of the guard period. The length of the cyclic prefix can also be minimized by using a channel equalizer.

We have the following:

$$v = T_g f_s,\ N = Tf_s,\ \text{or}\ f_s = \frac{N}{T} = \frac{v}{T_g}$$

The value f_s is the sampling frequency, and T_g is the guard period. The sampling frequency should be higher than twice that of the usable channel bandwidth.

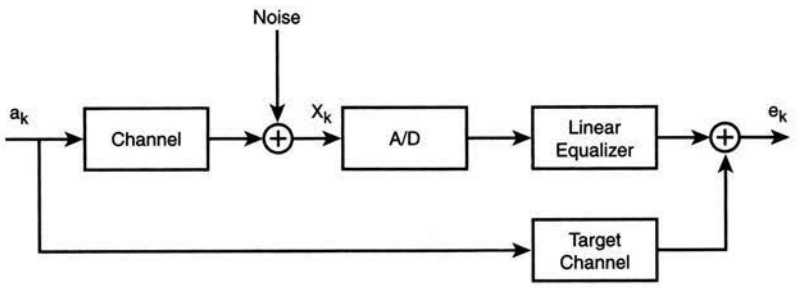
A time domain channel equalizer is necessary for the DMT receiver to reduce the amount of channel distortion present in the subscriber loop environment. The channel equalizer reduces the length of the cyclic prefix.

Note

In theory, a linear equalizer can reduce the cyclic prefix to zero length; however, a linear equalizer operating on the sampling rate to reach that goal might not be practical to implement in hardware. Furthermore, such a linear equalizer could cause excessive noise enhancement.

A linear equalizer with a small number of taps can reduce the cyclic prefix length to less than 30 sampling points. The conventional channel equalizer optimizes the desired signal to noise ratio at a particular decision point; however, the channel equalizer in general can modify the channel response according to particular needs just as what an FIR filter is capable of. A DMT Time Domain Equalizer can modify the loop channel response to the target channel response, as shown in Figure 13.11. The desired target channel response has the minimum mean square error when compared with any other equalized channel response.

Figure 13.11 Analysis of DMT Time-Domain Equalizer



The pair of DMT Time-Domain Equalizers and the target channel is similar to the pair of feedforward and feedback filters of a Decision Feedback channel Equalizer (DFE), except the impulse response of the target channel does not have to be causal. The optimal DMT Time-Domain Equalizer coefficients can be calculated using the same matrix operation method used for the calculation for those of a DFE, but in several iterations. The target channel impulse response is calculated only for evaluation, but not for real-time filtering. In practice, DMT Time-Domain Equalizer coefficients are identified iteratively in real-time in frequency or time domain.[91,92]

To complete the received signal detection, a Frequency-Domain Equalizer is also required. Although the inter-symbol interference has been mainly avoided by utilizing a cyclic prefix, amplitude attenuation and phase delay differences still exist for different subchannels. We can imagine each DMT baud pulse consisting of 256 sinusoids of different frequencies. At the transmit side, all these sinusoids are synchronized by the IFFT operation; however, these sinusoids are not synchronized in phase after the FFT operation at the receiver due to the channel distortion.

The purpose of the Frequency-Domain Equalizer (FEQ) is to correct phase and amplitude distortion of each subchannel such that a unified symbol detection mechanism can be applied to all subchannels. The FEQ has one complex tap for each subchannel resulting to an N/two-dimensional equalizer coefficient vector. The FEQ coefficient vector has the inverse frequency response of the equivalent channel, including transmit and receiver filters and the Time-Domain Equalizer. FEQ is performed by multiplying the FFT output with the FEQ coefficient vector.

Bit Loading Algorithm

Based on the Signal to Noise Ratio (SNR) of each subchannel, the number of bits that can be transmitted through each subchannel for certain error probability can be calculated as follows:

$$b_j = \log_2\left(1 + \frac{SNR_j}{\Gamma}\right)$$

The $SNRj$ is expressed in power ratio. Γ for an error rate of 10^{-7} is 9.8 dB or 9.55 in terms of power ratio. Γ of 9.8 dB is the reference point for uncoded system with 0 dB performance margin. To achieve an additional SNR performance margin, the value of Γ should be adjusted accordingly. For the same error rate, but with a 6 dB performance margin, the value of Γ would be increased to 15.8 dB or 38.02 in terms of power ratio. The value of Γ should be adjusted accordingly if any coding gain is to be considered. To add a coding gain of 3 dB, Γ will be reduced to 12.8 dB or 19.05 in terms of power ratio.

Note

It should be noted that Γ is only used here for analytical purpose—not as a design parameter.

The computation for the number of bits that can be transmitted through each subchannel will result in a fractional number of bits for each subchannel. An integer number of bits for up to 11 bits per subchannel is expected if a simple two-dimensional QAM-like constellation is to be implemented; however, the use of an integer number of bits could reduce the transmission efficiency.

One way to improve the transmission efficiency with integer number of bits is to assign a different signal power for each subchannel. The other way to improve transmission efficiency is to use a multidimensional signal constellation.[93] Up to 1 dB of performance improvement can be achieved by using these advanced loading algorithms.

The overall transceiver throughput is the sum of the number of bits for all subchannels times the modified baud rate, that is, the inverse of the baud period plus the guard period.

The computation of the performance margin for a target throughput can be achieved using the following expression in several iterations:

$$R = \sum_{j=1}^{256} b_j \frac{1}{T + T_g}$$

In other words, we first choose a particular SNR margin and calculate corresponding transmission throughput. We then adjust the value of the SNR margin and repeat the transmission throughput calculation until the desired transmission throughput is obtained.

Pilot Tone for Timing Recovery

The clock synchronization between a DMT transmitter and a DMT receiver is maintained through a timing recovery circuit in at least one transmission direction. This synchronization is based on the configuration where the transmitter and the receiver at one end of an ADSL system use their own clock source, and the transmitter and receiver at the other end use the recovered clock.

The timing recovery circuit of a DMT system involves the transmission of a pilot tone at the clock source end and the recovery of the pilot tone at the other end. The pilot tone could have periodic phase shift to reduce its interference with other tones. The transmit power of the pilot tone is usually a little lower than other information bearing subcarriers. The recovered pilot tone is usually filtered with a phase locked loop to increase its stability. The transmit power of the pilot tone may actually be more than the nominal transmit power of the information bearing subcarriers, especially for longer loops.

Note
A pilot tone is a subcarrier dedicated only for the purpose of timing recovery. A pilot tone has a fixed amplitude and fixed or alternating phases.

DMT ADSL Transceiver Parameters

The transceiver parameters[94] of a DMT system are summarized in Table 13.2 for the downstream channel.

Table 13.2 DMT Downstream Parameters

Symbol rate	4 kHz
FFT size	512 samples
Cyclic prefix	32 samples
Synchronization	Average 8 samples/symbol
Sampling rate	2.208 MHz
Transmit power	20 dBm
Time-Domain Equalizer	16 taps
Highpass filter (without echo cancellation)	6th order elliptic, passband corner frequency 62.5 kHz, 1 dB passband ripple

With the parameters in Table 13.2, the downstream DMT subcarriers are spaced at 4.3125 kHz intervals. The lowest carrier available is at 64.6875 kHz or 12.938 kHz with echo cancellation. The highest carrier available is at 1099.6875 kHz. The transceiver parameters of a DMT system are summarized in Table 13.3 for the upstream channel.

Table 13.3 DMT Upstream Parameters

Symbol rate	4 kHz
FFT size	64 samples
Cyclic prefix	4 samples
Synchronization	Average 1 sample/symbol
Sampling rate	276 kHz
Transmit power	7 dBm
Time-Domain Equalizer	32 taps
Lowpass filter	6th order elliptic, passband corner frequency 43.875 kHz, 1 dB passband ripple

With the parameters in Table 13.3, the upstream DMT subcarriers are also spaced at 4.3125 kHz intervals (identical to the downstream direction). The lowest carrier available is at 11.25 kHz (leaving below 10 kHz for POTS), the highest carrier available is at 42.75 kHz or 271.6875 kHz with echo cancellation. In addition, we need to meet an error rate of 10^{-7} and have a POTS highpass filter with a corner frequency of higher than 4 kHz. The performance should be examined with a background noise level of –140 dBm/Hz as well as NEXT and FEXT noise.

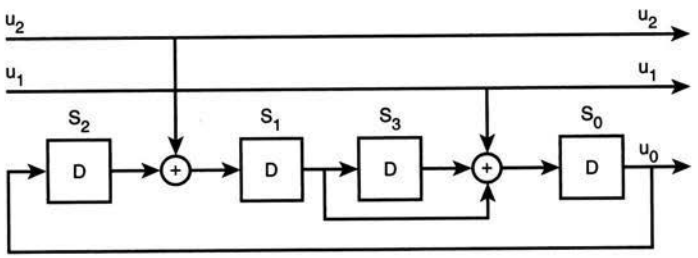
DMT ADSL Trellis Coding

The use of Trellis-Coded Modulation (TCM) is an option for ADSL to either meet performance margin requirements for long loops or increase the transmission throughput under a certain performance margin. Wei's 16-state 4D code, which has emerged from ten years of study and development, was proposed for ADSL.[95] This code has a theoretical gain of 4.5 dB and a practical gain of 4.2 dB against background and crosstalk noise.

In the TCM decoder, maximum-likelihood decoding using the Viterbi algorithm is employed to estimate the transmitted sequence. The use of a four-dimensional code (compared with a two-dimensional code) has the advantage of a smaller constellation expansion, and hence a smaller peak-to-average power ratio. As a result, it is more immune to nonlinear distortion. With the same decoding complexity, the four-dimensional codes also give higher coding gains.

The proposed Wei's code uses a rate 2/3 (2 bit/3 bit) convolutional code with parity check matrix (D3+D2, D, D4+1). The number of bits in a four-dimensional symbol increases from the uncoded case of b bits to the coded case of b+1 bits. The 2^{b+1} points are partitioned into eight cosets. Two of the b message bits are fed into the convolutional encoder. The three output bits from the convolutional encoder are used to specify the coset. The other b-2 message bits determine which point in the coset should be transmitted. Figure 13.12 shows the structure of the convolutional encoder.

Figure 13.12 Structure of the Convolution Encoder



The minimum square Euclidean distance between points in the same coset is 4, with 24 neighbors. The minimum square Euclidean distance between coded coset sequences is 6. The fundamental coding gain of the code is determined by the smaller of the two, and is equal to

$$10\log_{10} 4 - 1/2\ 10\ \log_{10} 2 = 4.52\ \text{dB}$$

The term $1/2\ 10\ \log_{10} 2$ is for the adjustment of constellation expansion per two dimensions. In the uncoded case, there are only eight nearest neighbors for each four-dimensional symbol. The factor of 3 increase in the number of nearest neighbors causes the coding gain to decrease from 4.52 dB to an effective value of 4.2 dB. Trellis coding and decoding are applied to each subcarrier individually.

DMT ADSL Forward Error Correction (FEC) Coding

To accommodate maximum flexibility for the providers and end users of ADSL services, it was recommended that FEC be selectively applied to 32 kbps increments of the composite data streams to, or from, the Central Office.[96] This permits FEC to be included or excluded selectively both on a call-by-call basis and within portions of the composite data stream.

As an example of mixed requirements for FEC in an ADSL service, consider transmitting a one-way DS1 stream from the Central Office to the remote terminal (CO to RT) along with a bi-directional basic rate ISDN channel. The end user may require high reliability on the one-way channel because the one-way channel might contain highly compressed data with no possibility for request for retransmission. For this portion of the ADSL signal, FEC is necessary. Voice services and duplex data services with their own embedded protocols will require minimum latency on the embedded ISDN channel. For this portion of the ADSL signal, FEC is optional.

FEC of ADSL employs Reed-Solomon codes based on symbols of 8 bits=1 byte (that is, a code over $GF(2^8)$). The maximum codeword length for such codes is 255, with varying error correction capabilities. Relatively long codeword lengths for a minimal redundancy of 8–10 percent were used. The redundancy and interleave depths chosen accomplish the following: correct for error bursts of up to 1/2 ms of transmitted data and reduce the background bit error rate (BER) from 10^{-7} to 10^{-10}.

Tables 13.4–13.6 summarize the chosen encoding system. Table 13.4 specifies interleave depths (the number of codewords in the interleave matrix) and codeword lengths (specified in ms) for uncoded data rates ranging from 32 kbps to 7584 kbps. Table 13.4 aids in the calculation of the number of message bytes in a codeword for any uncoded data rate. Note that the interleave depth times the codeword length, which equals the interleave latency, is kept constant at 16 ms.

Table 13.4 Interleave Depth and Codeword Length Parameters

Data Rate (kbps)	Interleave Depth (codewords)	Codeword Length (ms)
32–64	1	16
96–224	2	8
256–448	4	4
480–928	8	2
960–1888	16	1
1920–3776	32	0.5
3808–7584	64	0.25

Table 13.5 and Table 13.6 specify the number of check bytes in a codeword for data rates below and at or above 256 kbps, respectively. For the flexibility offered, very few parameters must be stored in the ADSL unit to implement selectable FEC.

Table 13.5 Message Bytes and Check Bytes (Data Rates < 224 kbps)

Data Rate (kbps)	Message Length (bytes)	Redundancy Length (bytes)
32	64	8
64	128	16
96–128	96–128	12
160–192	160–192	16
224	224	20

Table 13.6 Message Bytes and Check Bytes (Data Rates > 224 kbps)

Message Length (bytes)	Redundancy Length (bytes)
128–147	12
148–175	14
176–207	16
208–237	18

As a simple example in the use of Tables 13.4–13.6, consider adding FEC to an uncoded data rate of 1568 kbps. Using Table 13.4, we see one codeword contains (1568 kbits/sec)×(1 msec/codeword)×(1 byte/8 bits)=196 message bytes. Using Table 13.6, we find

that for a codeword of length 196 bytes, we need to add 16 check bytes to form a codeword of 212 bytes. The coded data rate, or line rate, is calculated as (212 bytes/codeword)×(1 codeword/1 msec)×(8 bits/byte)=1696 kbps.

Interleaving for Reed-Solomon Coding

The combination of Reed-Solomon code with interleaving is effective at correcting errors caused by impulse noise in the subscriber loop plant. Some studies indicate that a burst error correction capability of 500 microseconds (μs)=1/2 ms should provide sufficient protection against the effects of nearly all impulses encountered for a well-designed ADSL system.[97] On average, the time between impulses was found to be 67 ms.[98] For redundancies in the range of 7–9 percent, the required end-to-end latency is about 16 ms to correct for 500 μs of data with errors. To correct for 1000 μs of data with errors would require double the latency or double the redundancy.[99]

Consider an uncoded transmission rate of 1600 kbps=200 kbyte/s. Adding 7 percent redundancy, at this data rate equal to 14 kbytes/s, results in a coded transmission rate of 214 kbytes/s. The impulse causes a burst of 1/2 ms=107 bytes coded data destroyed. Using a t=7 byte (2t=14 check bytes) error correction capability FEC code would require an interleave depth of at least 16 codewords to correct this burst of 107 bytes of data with errors.

At a coded data rate of 214 kbytes/s, a codeword of length 214 bytes would last 1 ms. The latency for an interleave depth of 16 codewords is thus 16 ms. Conventional block interleaving would require 16 ms for interleaving at the encoder and an additional 16 ms for deinterleaving at the decoder, but convolutional interleaving, as described later in this section, reduces the total delay to 16 ms.

Using a fixed latency of 16 ms for any data rate, you can achieve the same burst error correction capability with smaller codewords and proportionally larger interleave depths. For example, relatively short constituent codewords are used in the concatenated Reed-Solomon codes for compact discs. Longer codewords, however, allow for additional redundancy to be added for random bit error protection with little additional cost. Increasing the number of check bytes in the previous example from the minimum 2t=14 to 2t=16 adds only 1 percent additional redundancy and results in a background bit error rate (BER) improvement from 10^{-7} to 10^{-10}. This last result is found using the following formula. Assume the following parameters:

- A burst error destroys x bytes per codeword
- A background random bit error rate (not due to the burst) of p

- A codeword length of N
- A redundancy of $2t$ bytes per codeword

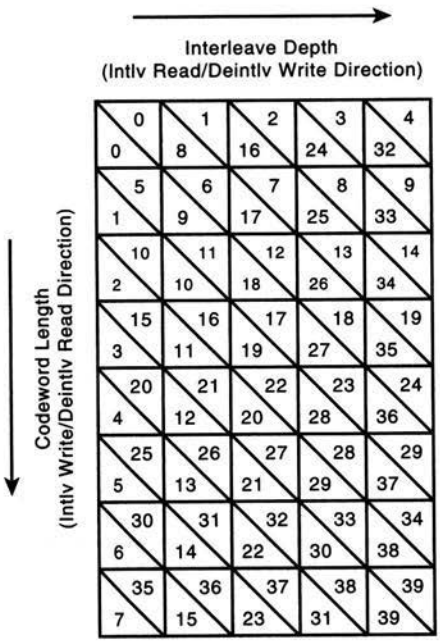
The bit error rate[100] after burst and random error correction is given approximately by

$$\frac{1}{2}\binom{N-x}{t-x+1}\frac{t+1}{N}(8p)^{t+1}\frac{1}{2}\binom{n-x}{t-x+1}\frac{t+1}{n}(8p)^{t+1}$$

Adding one byte error correction capability to that needed for burst error correction, we chose $t=x+1$. Substituting $p=10^{-7}$, you can confirm the background bit error rate is reduced to the order of 10^{-10} for the codeword lengths chosen.

Convolutional interleaving reduces the end-to-end latency incurred for burst error correction to half the latency incurred by block interleaving. This reduction is achieved by alternately reading and writing to the interleave matrix for each encoded byte. First, consider block interleaving. Figure 13.13 illustrates the block interleave matrix for a codeword length of N=8 bytes and an interleave depth d=5 codewords.[96]

Figure 13.13 Block Interleave/Deinterleave Matrix

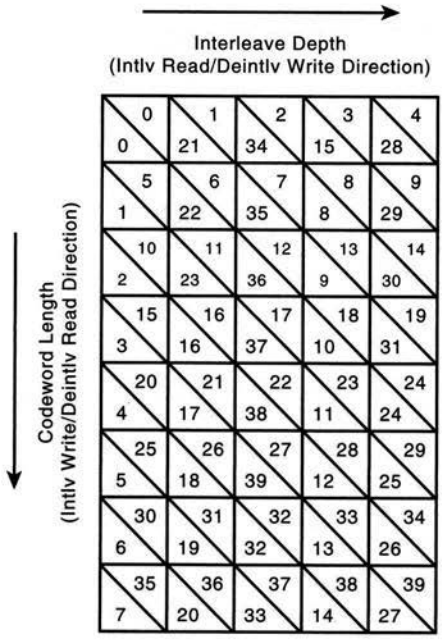


Block interleaving requires writing all positions in an entire interleave matrix prior to the first byte being read. The lower-left number x in each square in the matrix indicates the *time* index when writing into that location. The upper-right number y in each square in the matrix indicates the time index when reading from that location. Of course the entire matrix is written prior to any byte being read. Therefore, in actual implementation, two separate interleave matrices are used: one from which the current byte is read and one into which the current byte is written.

The same matrix describes deinterleaving, except the lower-left number in a square indicates the time index when reading from that location, while the upper-right number in a square indicates the time index when writing to that location. Deinterleaving adds a delay of 40+x–y bytes at the decoder end. The end-to-end delay with block interleaving for this example is thus 80 bytes. Note also that for correct decoding, the block deinterleaver must know the position of the beginning of the interleave matrix.

Now consider convolutional interleaving. As shown in Figure 13.14, in convolutional interleaving, after writing a byte into the interleave matrix, a previously written byte is read from the same interleave matrix. Only one interleave matrix is used for both reading and writing, so convolutional interleaving requires half the memory space of block interleaving.

Figure 13.14 Convolutional Interleave/Deinterleave Matrix



While the interleave matrix is read sequentially horizontally as with block interleaving, the matrix is written in a more complex manner, as shown by the numbers in the matrix. In general, the byte at the beginning of a codeword is written to, and read from, the same location. The remaining bytes in a codeword are written sequentially vertically within the same column as the first byte. For example, the first byte in a codeword experiences no delay (written and then read immediately thereafter).

At the decoder end, the same matrix describes deinterleaving except the bytes are now written sequentially horizontally and are now read in the more complex manner vertically. Thus, the first byte in a codeword incurs the minimum delay in the interleave matrix, but incurs maximum delay in the deinterleave matrix. The end-to-end total delay for this example is 40 bytes—exactly half the delay incurred by block interleaving.

Upon examining the interleave matrix, you should also note that the convolutional deinterleaver must know only the position of the beginning of a codeword to deinterleave correctly. Thus with convolutional interleaving, FEC synchronization need only search through a codeword depth rather than the entire interleave matrix. Note that for the entire interleave matrix to be used, the codeword length and interleave depth must be coprime. For ADSL, the interleave depths chosen are all powers of 2. Any odd codeword length will then be coprime to the interleave depth.

Convolutional interleaving incurs half the latency, requires half the memory, and needs less time for resynchronization compared with block interleaving. For this savings, convolutional interleaving requires a more complicated addressing scheme, needs alternate reading and writing of the interleave matrix, and requires an interleaving/deinterleaving codeword length coprime to the interleave depth. The convolution interleaving was selected for DMT ADSL.

DMT Asymmetrical Echo Cancellation

The asymmetrical echo cancellation technique can be used to improve the performance of an ADSL system. Although the upstream/downstream overlapped spectrum created by echo cancellation is NEXT limited, it still can provide a high enough SNR to be useful for transmission in both upstream and downstream directions. At the ADSL data rates, traditional time-domain and frequency-domain LMS echo cancellers can be prohibitively expensive to implement.

Mixed frequency and time-domain techniques are proposed to achieve low-complexity echo cancellation. The use of these techniques can achieve the required 70 dB of echo cancellation level.

A time-domain echo canceller needs to complete the discrete convolution process in the scale of the echo impulse response during every sampling interval. The echo impulse response could be as long as a few hundred sample points. For a sampling rate of about 1 MHz, a time-domain echo canceller would need a few hundred million multiplications and addition operations per second. On the other hand, if the echo cancellation is implemented in the frequency domain, assuming both echo path and data vectors are available in the frequency domain, the operations need to be completed only during each DMT baud interval. For 256 subcarriers and a baud rate of 4 kHz, the required operations are only a few million complex multiplication and addition operations per second.

For the mixed frequency and time-domain approach, most of the echo emulation is performed in the frequency domain. Because data vectors and therefore the echoes are not periodic, the time-domain convolution cannot be exactly duplicated in the frequency domain as a block multiplication. The circular nature of an echo block can be reinstalled by *tail cancellation* and *cyclic reconstruction*.[101,102]

Note

The tail cancellation is the subtraction of the tail caused by the previous data block, and the cyclic reconstruction is the replacement of a cyclic prefix corresponding to the current data block.

The combined process involves the convolution of an estimated echo path impulse response with portions of the current and the previous echo blocks.

Due to the asymmetrical transmission throughput nature, the sampling frequencies of transmission and receiving paths are different. Thus, a decimation, in the *Asymmetrical Terminal Unit-Central Office* (ATU-C), or an interpolation, in the ATU-R, process is necessary to match the rate of an analog to digital converter (ADC) to that of a digital to analog converter (DAC). Figure 13.15 shows a functional block diagram for an ATU-C DMT echo canceller. Figure 13.16 shows a functional block diagram for an ATU-R DMT echo canceller.

In the ATU-C, as illustrated in Figure 13.15, the ADSL upstream receiver has an FFT size of N. The ratio between the downstream DAC sampling rate and the upstream ADC sampling rate is k. Therefore, the ADSL downstream transmitter has an FFT size of kN. The echo has a tail of M sampling points in the transmitter sampling rate. The last $M-1$ previous and current data block points are combined through subtraction and used for tail cancellation and cyclic construction.

Figure 13.15 ATU-C DMT Echo Canceller

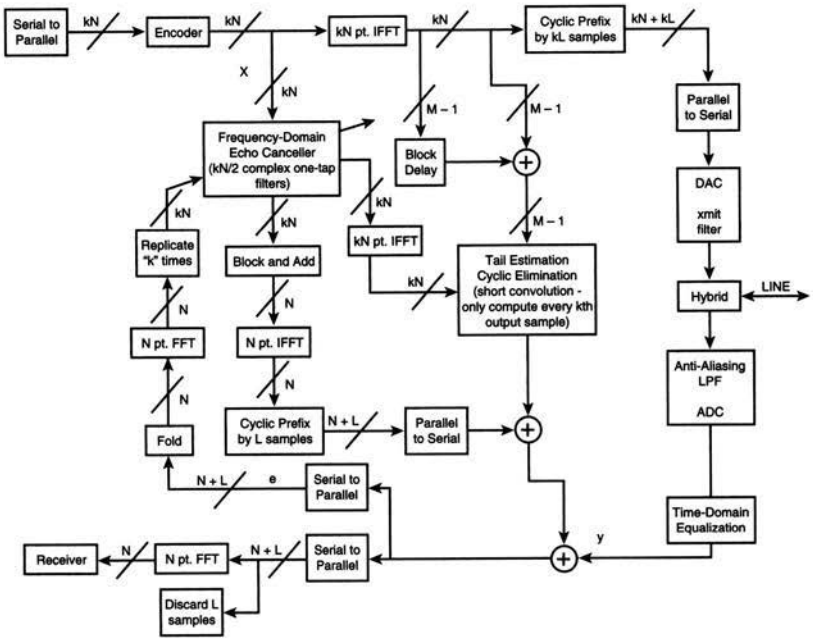
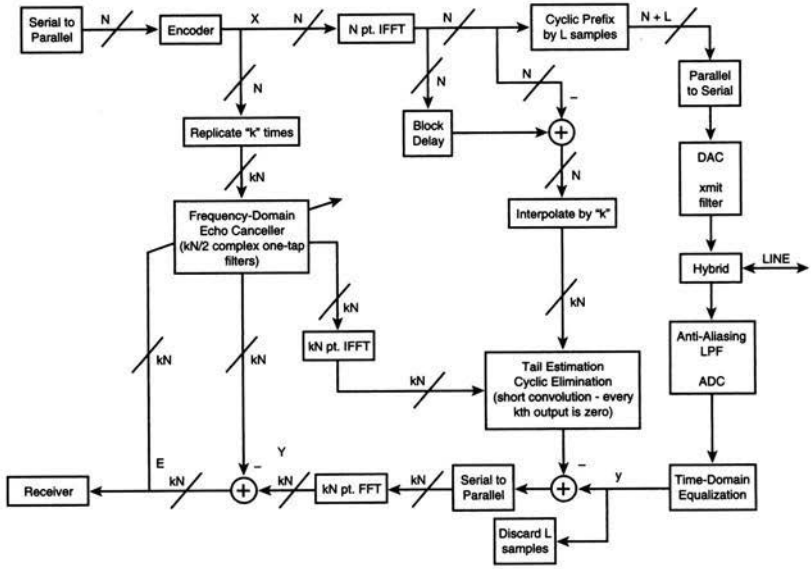


Figure 13.16 ATU-R DMT Echo Cancellation



The estimated frequency-domain echo path is also transformed into time domain for the tail cancellation and cyclic construction purpose. The convolution output for tail cancellation and cyclic construction is only needed at the receiver sampling rate. The main echo duplication is generated in the frequency domain, decimated by a factor of k, and converted into time domain for echo cancellation. The main echo replica is generated in the frequency domain, but used in the time domain to avoid the out-off-band aliasing effect.

In ATU-R, transmitting data in both the frequency domain and the time domain is interpreted to the receiver sampling rate. The interpolated frequency-domain data is used for main frequency-domain echo replica generation. The interpolated previous and current data block points in time domain are combined through subtraction and used for tail cancellation and cyclic construction. The estimated frequency-domain echo path is transformed into time domain for the tail cancellation and cyclic construction purpose. The main echo replica is generated in the frequency domain and combined with FFT transformed received data for direct frequency-domain echo cancellation.

DMT ADSL Transceiver Implementation

As for other DSL systems, only if a transceiver can be implemented with a low-cost VLSI (Very-Large-Scale-Integrated) circuits solution will ADSL find its broad acceptance. Major functions that an ADSL transceiver needs to accomplish in the real time include complex FFT/IFFT, Time-Domain Equalizer, Frequency-Domain Equalizer, echo canceller, Reed-Solomon forward error correction, Trellis coding, timing recovery, digital interfaces of up to seven channels, and OAM.

ATU-C/ATU-R Transceivers

ATU-C and ATU-R transceivers can be divided into three major functional blocks: *Digital Interfaces* (DI), *Digital Signal Processing* (DSP), and *Analog Interfaces* (AI).[103] Figure 13.17 shows such a decomposition for ATU-C. Figure 13.18 shows such a decomposition for ATU-R.

Figure 13.17 ATU-C Functional Blocks

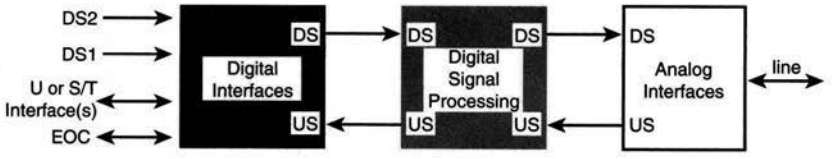
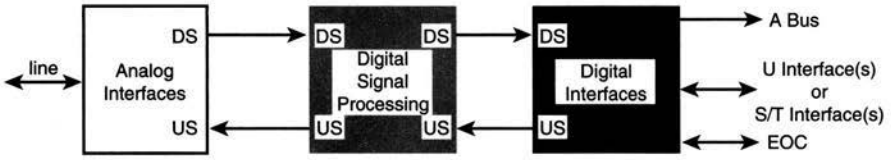
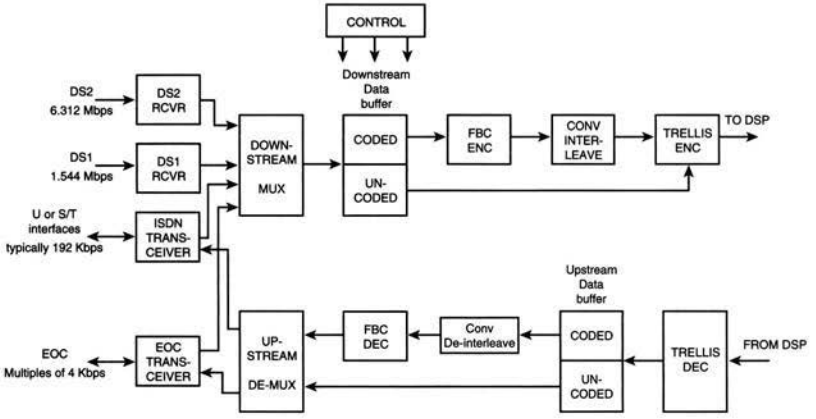


Figure 13.18 ATU-R Functional Blocks



The DI includes the acquisition and generation of data and clocks that interface the Central-Office switch or remote-terminal applications module. The DI essentially formats and unformats data for convenient processing by the DSP. Modulation and demodulation of data from and to the discrete time modulated transmission signals is performed in the DSP Subsection. The digital signals are converted and conditioned to and from the line in the AI. Figure 13.19 shows the functional blocks of ATU-C DI.

Figure 13.19 ATU-C DI Functional Blocks



In the downstream direction, high-speed primary data at DS1 (1.544 Mbps) or DS2 (6.312 Mbps) rates, low-speed secondary data at multiples of 16 Kbps (typically 192 Kbps), and EOC data at small multiples of 4 Kbps are received, processed, and transferred to the DSP. The processing includes multiplexing the various received data streams into one data stream, which is then buffered and separated into two types. The first type of data will be FEC coded and interleaved for error protection. The second type of data will not be interleaved but might be FEC encoded because of its sensitivity to extra latency. The coded, interleaved data, together with the non-interleaved data, is then Trellis encoded before they are passed to the DSP.

In the upstream direction, only the secondary data and the EOC data are present. The data from the DSP is first Trellis decoded, then separated into FEC coded and uncoded data through the upstream buffer. The FEC coded data is then deinterleaved and FEC decoded. This decoded data, together with the uncoded data, are demultiplexed into the secondary data and the EOC data, which are then transferred to the transceivers. Table 13.7 summarizes the detailed semiconductor requirements:.

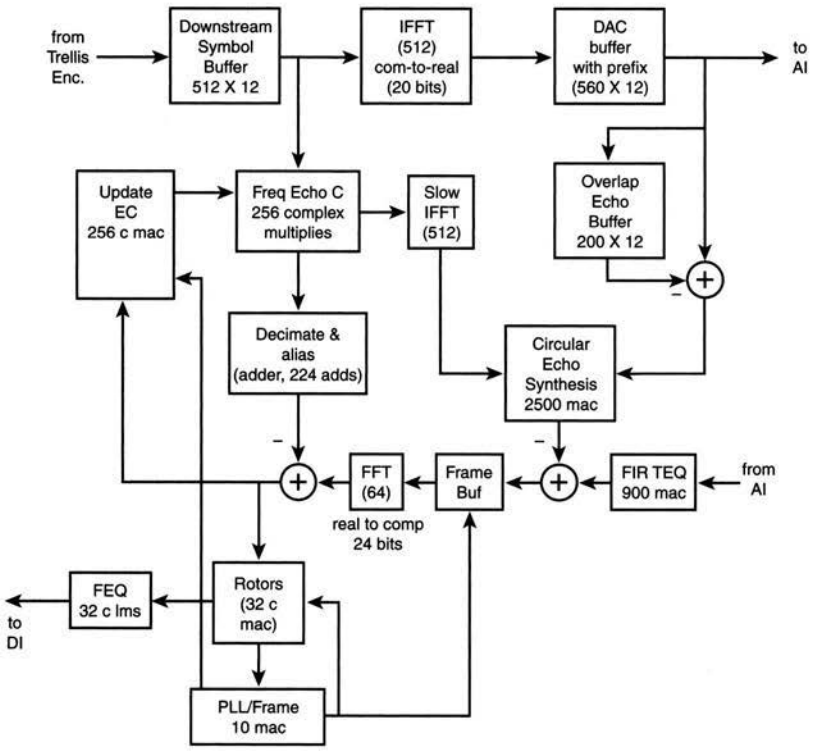
Table 13.7 DI Complexity Estimates

Functional Block	Random Logic	RAM	ROM
DS1 receiver	2000	N/A	N/A
DS2 receiver	2000	N/A	N/A
ISDN transceiver	2000	N/A	N/A
EOC transceiver	2000	N/A	N/A
Downstream multiplexer	1000	16	N/A
Downstream buffer	1000	512	N/A
FEC encoder	2000	18	128
Convolutional interleaver	1000	16 K	N/A
Trellis encoder	3000	192	1 K
Trellis decoder	15000	904	800
Upstream buffer	1000	512	N/A
Convolutional deinterleaver	1000	512	N/A
FEC decoder	7000	256	N/A
Upstream demultiplexer	1000	256	N/A
Control	8000	N/A	4 K×8
Total	49000	20 K	6 K

Figure 13.20 illustrates the ATU-C DSP functional blocks.

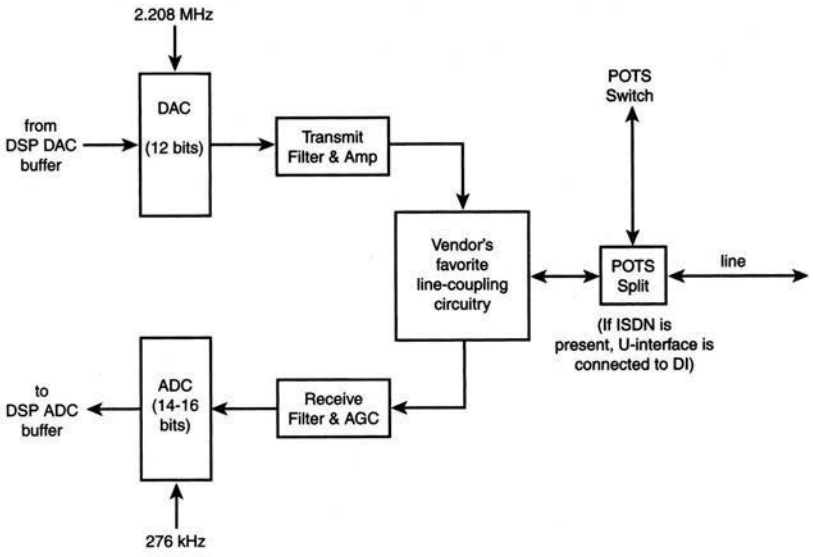
ATU-C DSP Functional Blocks Data symbols for each subchannel input are supplied by the Trellis encoder of the DI in 6-bit real and 6-bit imaginary parts. The downstream data symbols are modulated digitally by the 512 complex-to-real IFFT, and a 32-sample prefix is added to each block of time-domain samples at the DAC buffer, which is output to the AI at 2.208 Msamples/s.

Figure 13.20 ATU-C DSP Functional Blocks



The echo canceller uses an overlap method to make the echo appear periodic so that a simple 1-tap per subchannel echo canceller will provide excellent cancellation range. The upstream ADC samples at 276 kHz are filtered by an FIR TEQ and followed by the circular echo adjustment, frame buffering, and demodulation by the 64-point real-to-complex FFT. A decimated frequency-domain echo is subtracted, timing is digitally interpolated in the frequency domain to the remote clock, and a 1-tap LMS gain/phase adjustment (FEQ) is made to each subchannel so that common decision boundaries can be used by the Trellis decoder in the DI. The ATU-C DSP requires about 67,000 gates, 12 K RAM, 6 K ROM, and 67 MIPS (Millions of Instructions Per Second). Figure 13.21 illustrates the ATU-C AI functional blocks.

Figure 13.21 ATU-C AI Functional Blocks



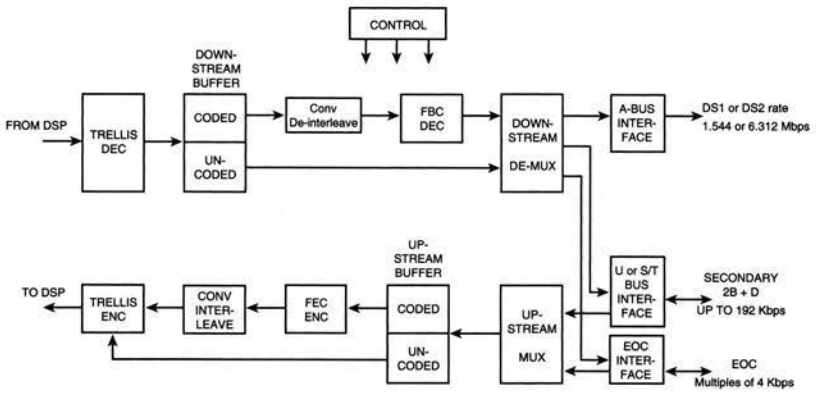
DAC precision is 12 bits at a sampling rate of 2.208 MHz, while ADC precision is 14–16 bits (good hybrids require 14, less sophisticated hybrids require 16) at a sampling rate of 276 kHz. The ADC clock is exactly eight times slower than the DAC clock, and both are run asynchronously with the CO clocks at a stable jitter-free crystal oscillator frequency. The jitter-free clocks are required for proper operation of the echo canceller in the DSP section. Interpolation and elastic queues in the DSP and DI sections realign clocks.

The ATU-R DI is very similar to the ATU-C DI. The main difference is that the decoder portion now runs at the downstream high data rate, while the encoder portion runs at the upstream low data rate. To save design effort, the basic computation modules (FEC encoder, FEC decoder, Trellis encoder, and Trellis decoder) are designed for both ATU-C and ATU-R DIs.

Figure 13.22 illustrates the functional blocks of an ATU-R DI.

In the upstream, low-speed secondary data at multiples of 16 Kbps (192 Kbps for example), and EOC data at small multiples of 4 Kbps are received, processed, and transferred to the Digital Signal Processor. The secondary data and EOC data are multiplexed, buffered, and separated into two types. The first type of data will be FEC coded and interleaved for error protection. The second type of data will not be interleaved because of its sensitivity to extra latency. The coded, interleaved data, together with the uncoded data, is then Trellis coded before they are passed to the DSP.

Figure 13.22 ATU-R DI Functional Blocks



In the downstream direction, in addition to the secondary and EOC data, primary data at DS1 (1.544 Mbps) or DS2 (6.312 Mbps) rates are also present. The data from the DSP is first Trellis decoded, then separated into FEC coded and uncoded data through the downstream buffer. The FEC coded data is then deinterleaved and FEC decoded. This decoded data, together with the uncoded data, are demultiplexed into the primary and secondary data and the EOC data. The ATU-R DI also requires about 49,000 gates, 20 K RAM, and 6 K ROM. Figure 13.23 illustrates the functional blocks of an ATU-R DSP.

Data symbols for each subchannel input are supplied by the Trellis encoder of the DI in 6-bit real and 6-bit imaginary parts. The upstream data symbols are modulated digitally by the 64 complex-to-real IFFT and a 4-sample prefix is added to each block of time-domain samples at the DAC buffer, which is output to the AI at 276 ksamples/s. The echo canceller uses an overlap method to make the echo appear periodic so that a simple 1-tap per subchannel echo canceller will provide an excellent cancellation range.

The downstream ADC samples at 2.208 MHz are filtered by a 16-tap FIR TEQ and followed by the circular echo adjustment, frame buffering, and demodulation by the 512-point real-to-complex FFT. An interpolated polyphase frequency-domain echo is subtracted, timing is digitally interpolated in the frequency domain to the Central Office clock, and a 1-tap LMS gain/phase adjustment (FEQ) is made to each subchannel so that common decision boundaries can be used by the Trellis decoder in the DI. The ATU-R DSP requires about 72,000 gates, 12 K RAM, 4 K ROM, and 132 MIPS. Figure 13.24 illustrates the functional blocks of an ATU-R AI.

Figure 13.23 ATU-R Digital Signal Processing Functional Blocks

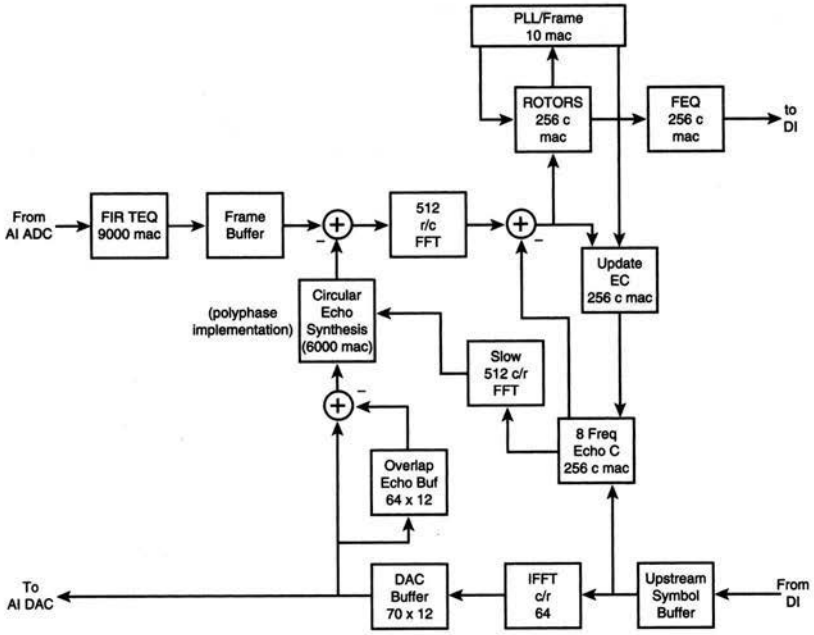
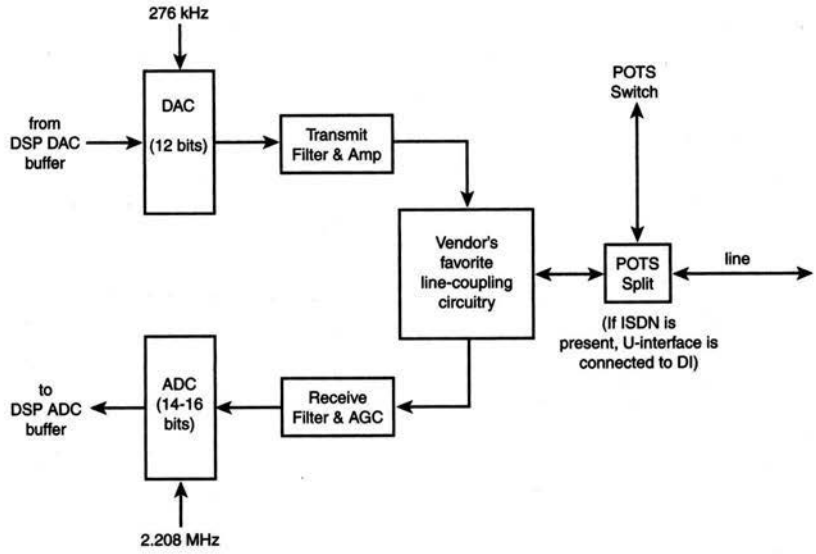


Figure 13.24 ATU-R AI Functional Blocks



DAC precision is 12 bits at 276 kHz, while ADC precision is 14–16 bits (good hybrids require 14, less sophisticated hybrids require 16) at 2.208 MHz. The ADC clock is exactly eight times faster than the DAC clock and both are run asynchronously with the ATU-R data clocks at a stable jitter-free crystal oscillator frequency. The jitter-free clocks are required for proper operation of the echo canceller in the DSP section. Interpolation and elastic queues in the DSP and DI sections realign clocks.

13.2.2 CAP Line Code

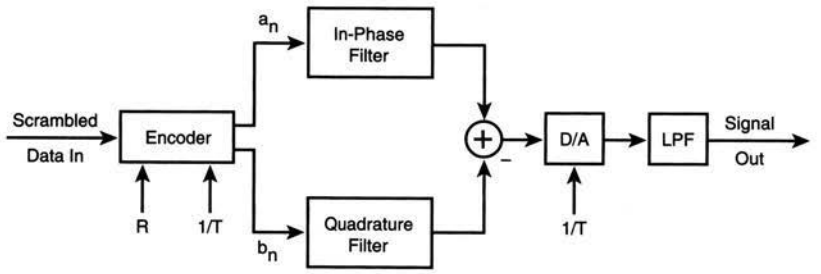
A CAP (Carrierless AM/PM) ADSL system consists of an upstream channel and a downstream channel. There is a transmitter and receiver pair for each channel and for each transceiver. A CO-side CAP transceiver, ATU-C, has a downstream transmitter and an upstream receiver. A residential or remote side transceiver, ATU-R, has an upstream transmitter and a downstream receiver.

CAP Transmitter Structure and Functionality

A CAP transmitter should have a scrambler, an encoder, a pair of transmitter filters, a Digital to Analog converter (DAC), and a lowpass filter.[104] The encoder converts the scrambled data into an in-phase channel data and a quadrature channel data. The transmitter filters convert the data streams into passband line signal. The DAC puts the filtered signal into analog format. The lowpass after the DAC removes harmonic images of the transmit signal. The encoded signal arrives at the baud rate.

The filters operate three to four times the baud rate. The DAC usually has a higher internal operating rate. The lowpass filter could be in analog format. A transmitter clock, locked to network incoming data or to recovered data by the receiver, should be available to functional blocks of the transmitter. Figure 13.25 shows the structure of a CAP ADSL transmitter.

Figure 13.25 A CAP Transmitter



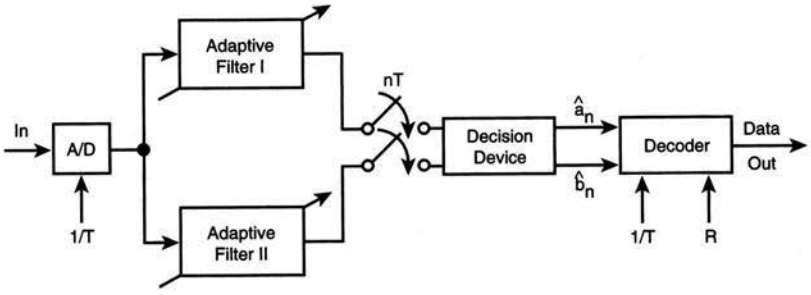
a) Transmitter Structure

CAP Receiver Structure and Functionality

A CAP receiver should have a clock recovery or timing generation circuit, an Analog to Digital Converter (ADC), a pair of adaptive channel equalizers, a symbol detection device, and a decoder. The clock/timing circuit is a very important part of the receiver and it can be independent or dependent of the receiver data path. The ADC transforms the received signal in digital format for further processing.

The data output rate of the ADC should be compatible with the operation rate of the pair of channel equalizers. The adaptive channel equalizers compensate for the signal distortion caused by channel intersymbol interference. The adaptive equalizers operate at a few times of the baud rate. The data output rate of the equalizer is the baud rate. The decision device converts the equalized signal to data symbols. The decoder converts the data symbol to the original bit stream. Figure 13.26 shows the structure of a CAP ADSL receiver with linear equalizers.

Figure 13.26 A CAP Receiver with Linear Equalizers



b) Linear Receiver Structure

The adaptive filters could be linear or decision feedback. The linear adaptive filters can also be used in conjunction with Tomlinson prefilters in the transmitter. The Tomlinson filter coefficients are usually identified with a decision feedback equalizer structure. To improve system performance, a DFE can be used instead of the linear equalizer in a CAP receiver. Figure 13.27 shows a CAP receiver with a conventional DFE.

The two feedforward filters of the CAP receiver, as shown in Figure 13.27, typically operate at a sampling rate of 3/T or 4/T. Figure 13.28 shows a noise predictive DFE structure. The noise predictive CAP DFE might have a slight performance edge compared to that of a conventional CAP DFE.

Figure 13.27 A CAP Receiver with Traditional DFE

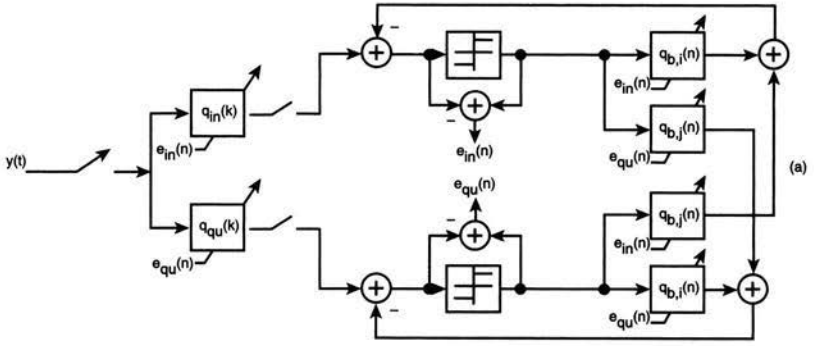
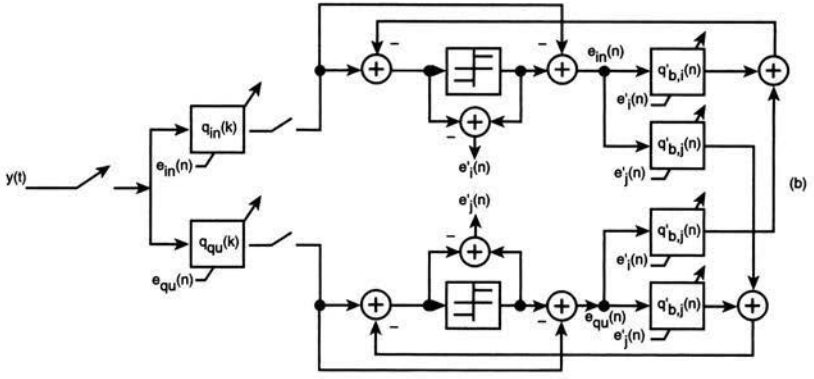


Figure 13.28 A CAP Receiver with Noise Predictive DFE



CAP Precoding Techniques

The Trellis coding does not work very well in conjunction with a DFE with long feedback filters. The DFE needs estimated signal symbols right after every decision point to drive the feedback filter. The Trellis coding, however, would not make a good decision until the probability of one possible tree is much higher than other possible alternatives. The decision needs to be made based on a past few baud intervals. Therefore, the correct decisions from a Trellis decode would be too late to be used by the feedback filter.

The Tomlinson precode approach[105] can be used to combine benefits of DFE and Trellis coding together. The idea is to move the feedback filter coefficients into the transmitter and prefilter the data symbols prior to transmission, therefore making the decision delay irrelevant. Because the modulation and the demodulation processes are transparent for the equivalent channel, the position of the feedback filter can be at either the transmitter or the receiver for a linear system.

One issue preventing the realization of this prefiltering approach is that the feedback filter used as a recursive filter in the transmitter might not be stable. Tomlinson proposed to use a modulo operation in the transmitter to stabilize the recursive feedback filter. Another modulo operation was also proposed in the receiver to recover the original data symbols. Figures 13.29 and 13.30 show a simplified DFE-based transceiver system and its Tomlinson precoding equivalent system, respectively.

Figure 13.29 A Simplified DFE System

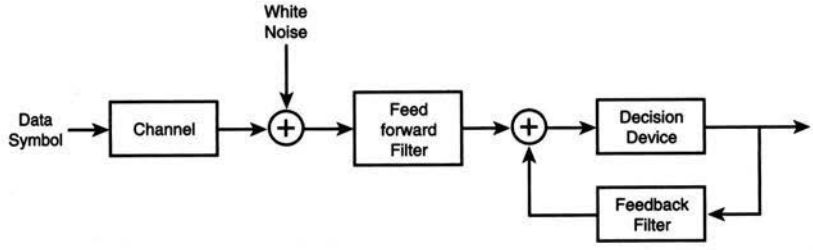


Figure 13.30 A Simplified Tomlinson Precoding System

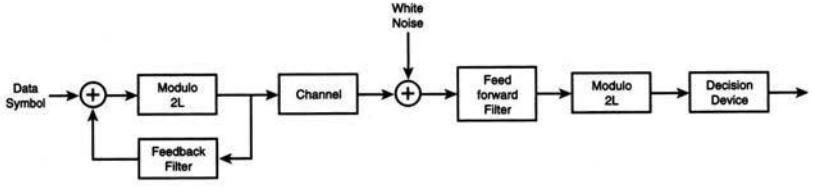
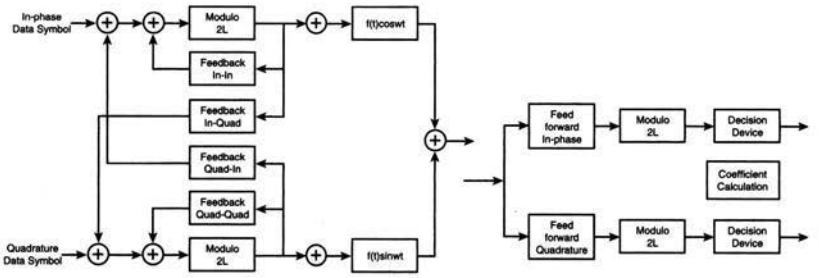


Figure 13.31 shows the transceiver structure for combining the CAP line code and the Tomlinson precoder together.

There are four modulo operations all together for both in-phase and quadrature channels. There are four feedback filters in the CAP transmitter. Filter coefficients of feedforward and feedback filters are calculated in the initialization phase. Feedback filter coefficients are

transmitted from the transmitter to the receiver. At the minimum, feedforward filter coefficients need to be adaptively updated during the normal transmission phase.

Figure 13.31 CAP Transmitter with a Tomlinson Precoder



CAP Transceiver Characteristics and Timing Recovery

The CAP transceiver is derived from the traditional QAM transceiver. Transceivers at both ends of a QAM system need to be synchronized to make the channel equalizer and the rest of circuits work properly. The synchronization of a QAM system is achieved through carrier recovery and timing recovery circuits.

The carrier recovery of a QAM system is necessary for the demodulation process to generate a baseband signal and to maintain the orthogonality between in-phase and quadrature channels. In a digital implementation of a QAM system, the carrier frequency can be chosen to have a fixed relationship with the signaling baud rate. The synchronization between carrier frequency and baud rate can simplify the implementation of carrier recovery and timing recovery circuits.

For the digital implementation approach of a CAP transmitter, the center (carrier) frequency of a CAP system is always synchronized with the signaling baud rate. The orthogonality between in-phase and quadrature channels of a CAP system can be maintained through passband channel equalizers and the received signal is directly recovered by these passband equalizers. Therefore, the carrier recovery circuit is not necessary for a CAP system. A good timing recovery circuit is sufficient for a CAP receiver to synchronize with the CAP transmitter. Because CAP passband equalizers are fractionally spaced compared with the baud rate, the requirement for the accuracy of the timing phase of the timing recovery circuits can be relaxed.

Based on received signal without pilot tone, we can optimize the sampling timing phase based on the maximum likelihood estimation procedure. Through analysis, we can obtain the likelihood expression as follows:

$$l\left(\tilde{\tau}\right) \sim \sum_{n=0}^{N} {q_n}^2\left(\tilde{\tau}\right)$$

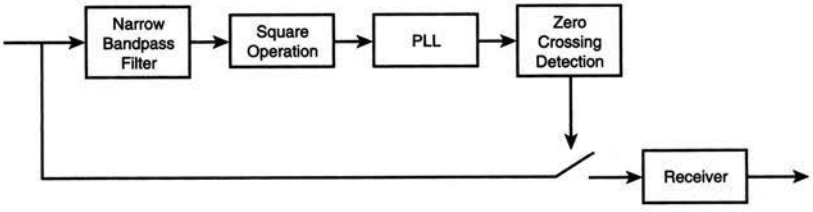
The values are as follows:

$q_n(\tilde{\tau})$ is the received signal after passing through a matched filter.

$\tilde{\tau}$ is the timing phase.

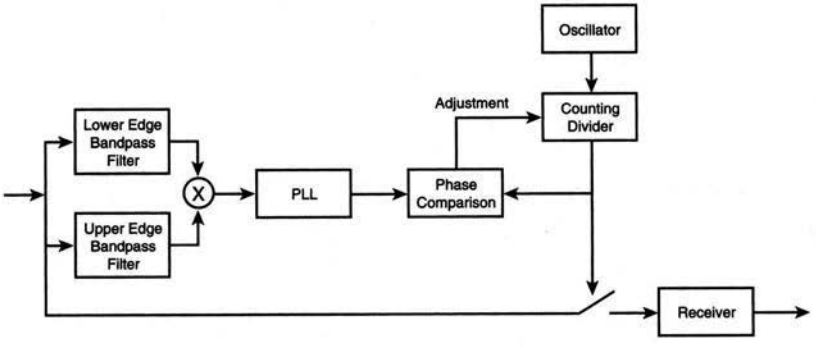
For a baseband system, the timing phase can be optimized through the practical circuit shown in Figure 13.32.

Figure 13.32 Square-Based Timing Recovery Circuit



The narrow bandpass filter reduces the timing jitter, the squarer operation generates desired tones, and the Phase Locked Loop (PLL) selects the desired tone at 1/2T Hz. This circuit relies on the energy in the excess band (right outside of the signaling band) to make it work. Figure 13.33 shows a similar timing recovery circuit for a CAP transceiver.

Figure 13.33 CAP Timing Recovery Circuit



Instead of the squarer operation, the energy at the lower band edge and at the upper band edge is multiplied to generate desired tones. The timing information recovered by the PLL is compared with that of the local clock which drives the sampling circuit. If the local timing phase is tardy, the value in the counting divider is reduced to slightly increase the timing clock frequency; otherwise, the value is increased.

CAP ADSL Transceiver Parameters

Table 13.8 summarizes the transceiver parameters[106] of a CAP system. With the upstream channel transmit spectrum above 90 kHz, this CAP ADSL system can be deployed with either POTS or ISDN Basic Rate Access services. Transmit spectrum preshaping at 3 dB is implemented for this CAP ADSL also, resulting in power density levels of about –37 dBm/Hz at around 170 kHz and –40 dBm/Hz at around 410 kHz.

Table 13.8 CAP Transceiver Parameters

	Downstream	Upstream
Constellation size	64 CAP	16 CAP
Baud rate	266.667 kHz	6 kHz
Throughput	1.6 Mbps	24 kbps
Sampling rate	1.066667 MHz	
Transmit power	12.1 dBm (100 ohms)	–4.8 dBm (135 ohms)
Transmit spectrum	170 kHz–410 kHz	96 kHz–102 kHz

CAP Transmitter Filter and Equalizer Coefficients

The transmitter filters are usually designed based on the raised cosine or square root raised cosine function. The raised cosine function in the time domain is[104]

$$g(t) = \frac{\sin \pi t / T}{\pi t / T} \frac{\cos \alpha \pi t / T}{1 - (2\alpha t / T)^2}$$

The transfer function in the frequency domain is

$$G(f) = \begin{cases} \frac{T}{2} - \frac{T}{2} \sin \frac{T}{\alpha} \left(f - \frac{1}{2T} \right) & 0 \le |f| \le \frac{1}{2T}(1-\alpha) \\ \frac{1}{2T}(1-\alpha) < |f| \le \frac{1}{2T}(1+\alpha) \end{cases}$$

The square root raised cosine function in the time domain is

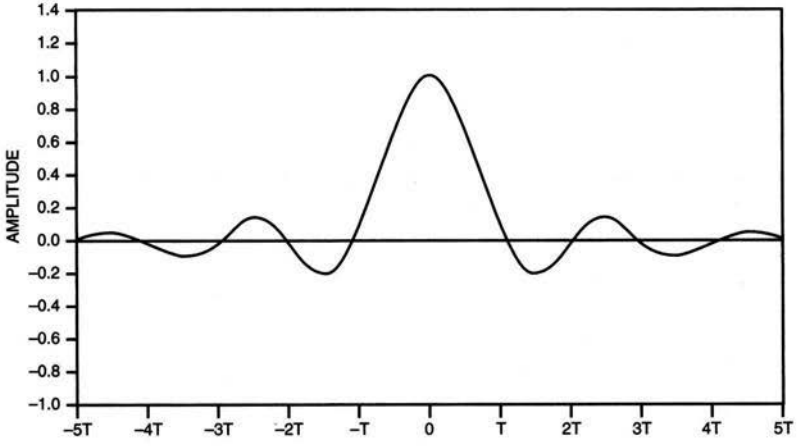
$$g(t)=\frac{\sin\left[\pi(1-\alpha)t'\right]+4\alpha t'\cos\left[\pi(1+\alpha)t'\right]}{\pi t'\left[1-(4\alpha t')^2\right]}$$

The value $t'=t/T$ and its transfer function is

$$G(f)=\begin{cases}\dfrac{T}{\sqrt{2}}\sqrt{1-\sin\dfrac{\pi T}{\alpha}\left(f-\dfrac{1}{2T}\right)} & 0\le|f|\le\dfrac{1}{2T}(1-\alpha)\\ & \dfrac{1}{2T}(1-\alpha)<|f|\le\dfrac{1}{2T}(1+\alpha)\end{cases}$$

Figure 13.34 shows the time domain raised cosine function.

Figure 13.34 Raised Cosine Function



Figures 13.35 and 13.36 show the in-phase and quadrature transmitter filters, respectively, based on the raised cosine function.

Due to the dual functionality of equalization and baseband signal recovery, the linear channel equalizer still needs to cover both *pre-cursor* (before the peak) and *post-cursor* (after the peak) of the channel impulse response. Feedforward filter coefficient values are a little different compared with those of linear equalizers, due to the addition of the feedback filter. Figure 13.37 shows feedforward filter transfer functions for conventional DFE and noise

predictive DFE (NPDFE). Figures 13.38 and 13.39 show corresponding real and imaginary feedback filter coefficients, respectively.

Figure 13.35 In-Phase Transmitter Filter

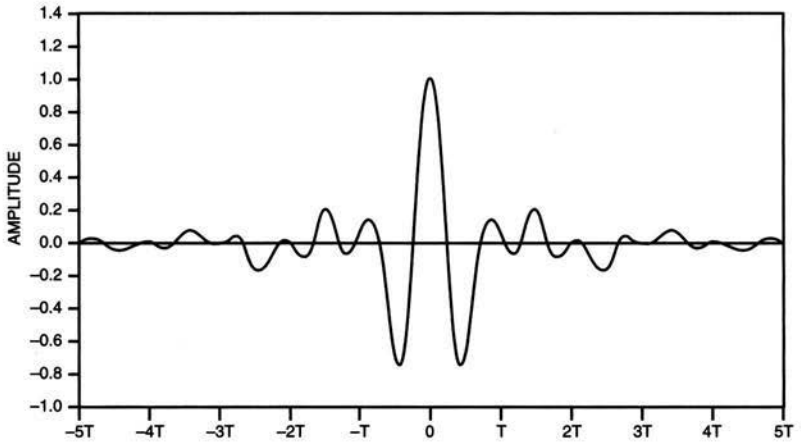


Figure 13.36 Quadrature Transmitter Filter

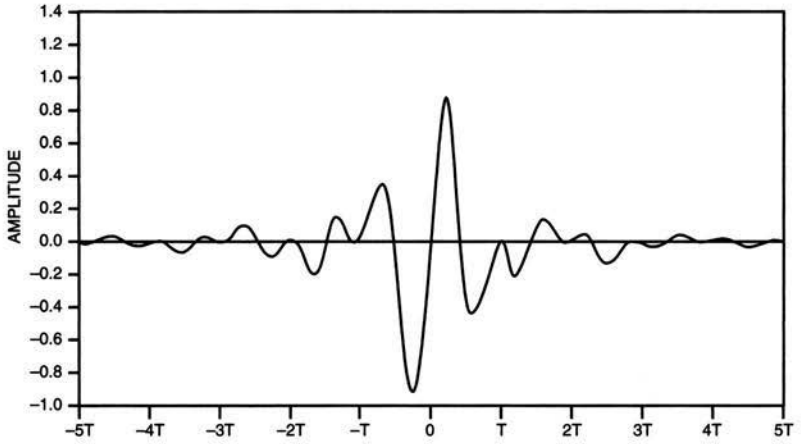


Figure 13.37 Transfer Function of Feedforward Filters

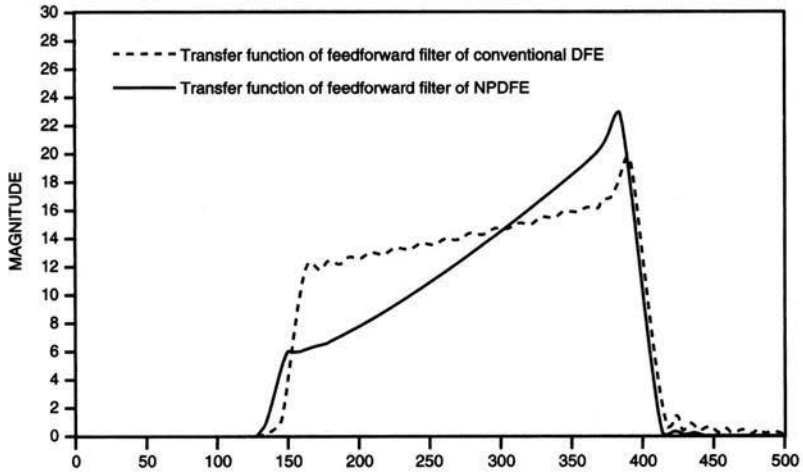
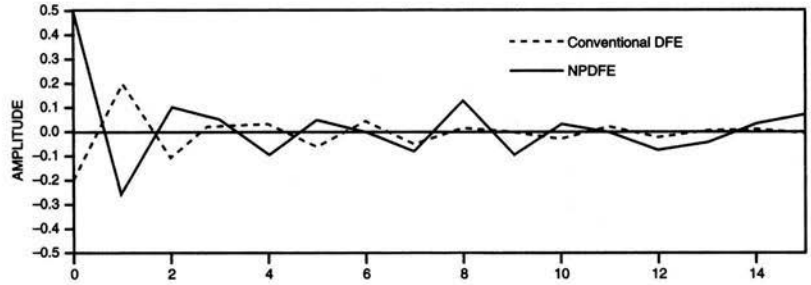


Figure 13.38 Real Feedback Filter Coefficients



CAP Implementation on the VDT and GAFF Chipset

A CAP ADSL transceiver can be implemented with the General Adaptive FIR Filter (GAFF) chip.[107] A GAFF can provide the signal processing functions required for ADSL operations that include passband channel equalization and Tomlinson precoding. In addition to FIR operations, the GAFF chip also incorporates binary functions. Figure 13.40 depicts the system architecture using three GAFF chips in conjunction with a VDT (Viterbi Decoder, Data Interface, and Timing Recovery) chip, an analog front-end functional block, and a general-purpose DSP for an ADSL transceiver.

Figure 13.39 Imaginary Feedback Filter Coefficients

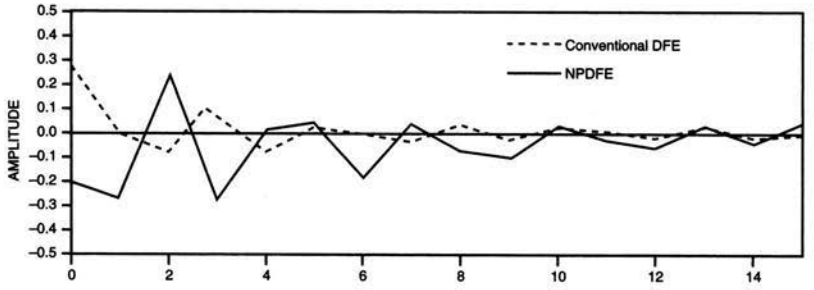
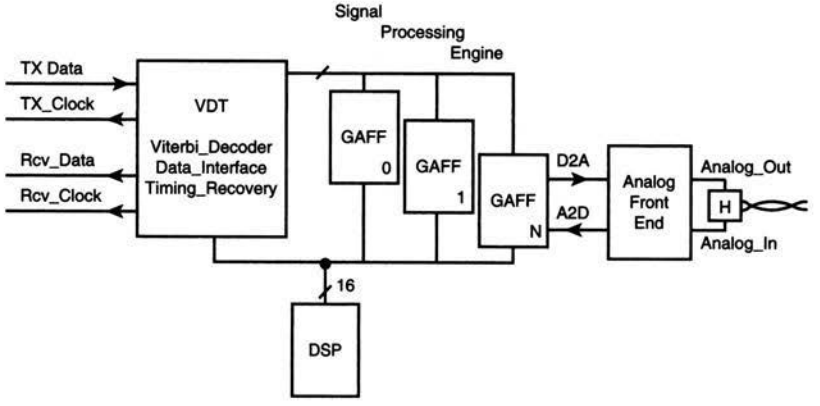


Figure 13.40 System Architecture for ADSL

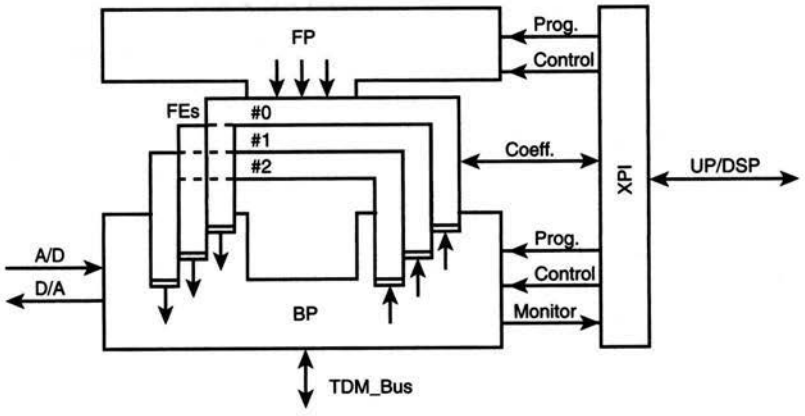


Three GAFF chips reside on a time division-multiplexed (TDM) bus, enabling the software to reconfigure system functions and interconnects. The GAFF chip is designed as a RAM-based, dual-processor engine consisting of an FIR processor and a binary processor. The FIR processor is a single-instruction, multiple-data processor. The binary processor is a simpler single controller with binary operations suitable for modem applications. Figure 13.41 shows the GAFF internal architecture.

The GAFF chip is composed of four circuit blocks:

- FIR processor (FP)
- FIR engines (FEs)
- binary processor (BP)
- external processors interface (XPI)

Figure 13.41 GAFF Functional Block Diagram



The FP contains a 128×32b program RAM that defines the signal processing operations. In addition to controlling its own program counter, the FP also controls the three FIR engines (FEs). The FP generates the coefficients and data address pointers, defines the signal-processing operations, and transfers results from the FEs to the output registers in the binary processor space.

The three FEs perform all operations in parallel, executing vector multiplications and coefficient updates concurrently. Each FE contains a multiplier, an accumulator, data multiplexors, and its own coefficients and data RAMS. The three FEs differ only by their data and coefficient values, not by the operations.

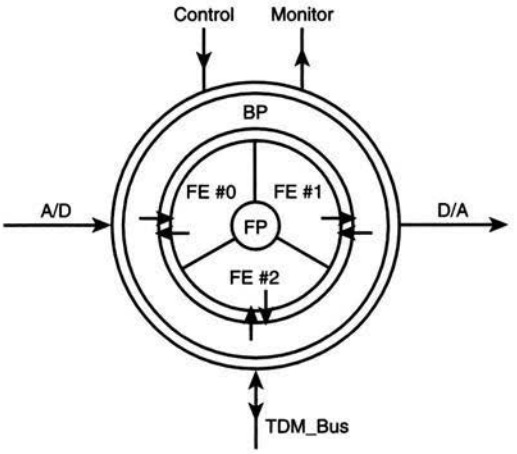
The binary processor (BP) is also fully programmable using a 128×32b program RAM that configures the FIR structure from the FEs' partial products. The BP performs basic binary operations (such as slicing) and defines system interconnectivity on the TDM Bus.

The external processor interface (XPI) supports the external bus interface to either a DSP or a microprocessor. The XPI permits code downloading, chip reconfiguration, and real-time control and monitoring.

The FP and BP are fully independent processors that are loosely coupled by input and output FIFO stages. The FP and BP program memories define the data and coefficient structures and their operations. The BP defines the input sources of the FEs, the output functions, and the error controls, while the FP defines the filter structures, the filtering operation, and the FIR outputs.

Each of the two program words is divided into subfields, enabling multiple operations (such as loop branch) to be performed in addition to control operations without any real-time penalties. Figure 13.42 shows the GAFF software model, illustrating data flow between the two processors and the external world. Many filter combinations are possible by independently programming the input content, filter structure, output functions, and error formats.

Figure 13.42 GAFF Software Model



The 37.3 mm² device is packaged in an 84-pin PLCC and dissipates 750 mW at 33 MHz. The chip is designed in a 0.9 μm double-metal Complementary Metal Oxide Semiconductor (CMOS) technology, with a mixture of custom layout macros and the standard cell technology, for a total of 240 k transistors. The multiply/accumulate section is custom designed for minimum area and maximum speed. The six custom RAMs for data, coefficients, and programs are also designed using area-efficient custom-layout tools. The custom cells are integrated into a high-speed standard-cell library.

At 33 MHz, the FP executes 100 M multiply/accumulate operations per second, while the BP section executes an additional 33 Mops, in addition to other program control operations that are executed in parallel. At a symbol rate of 1 Mbaud, the device can support a 99-tap transmitter filter.

13.3 *Performance Objectives for Mid-CSA Test Loops*

The 1.544 Mbps ADSL was proposed for resistance design loops. Test loops selected for DSL, as shown in Figures 11.30 through 11.35 in Chapter 11, "DSL (Digital Subscriber Lines)," are also used for simulation study and testing of this low throughput ADSL. CSA test loops, as shown in Figures 12.22 and 12.23 in Chapter 12, "HDSL (High-Bit-Rate Digital Subscriber Lines)," proposed for HDSL are used for simulation study and testing of high transmission throughput, 7 Mbps in the downstream ADSL. To verify the performance of the high throughput ADSL under T1 NEXT, six mid-CSA loops were also proposed.[108] Figure 13.43 shows configurations of these mid-CSA loops.

Figure 13.43 Mid-CSA Test Loops

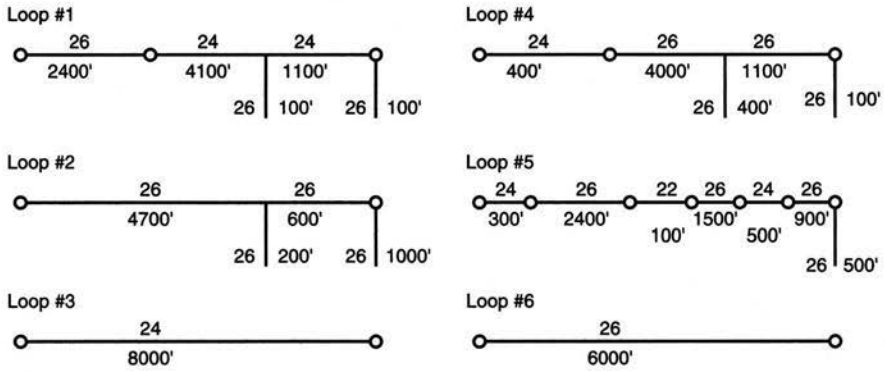


Figure 13.44 shows transfer functions of these mid-CSA test loops.

Figure 13.44 Transfer Functions of Mid-CSA Loops

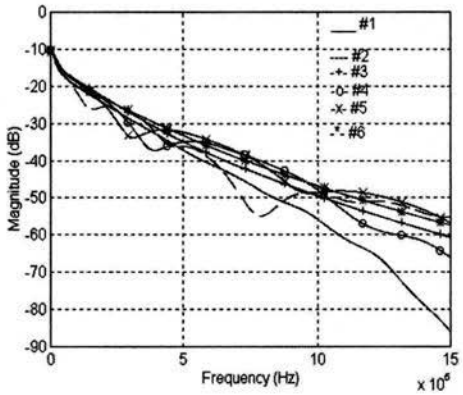
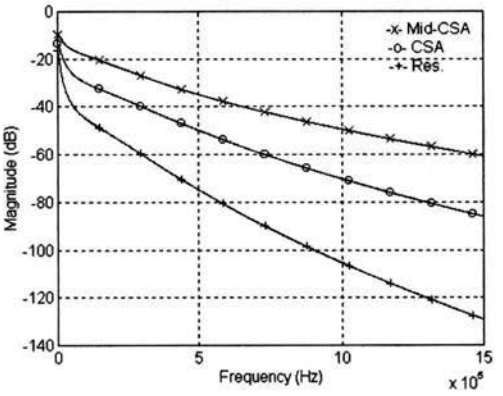


Figure 13.45 shows the transfer function of typical resistance design, CSA, and mid-CSA loops. The top mid-CSA loop has the least attenuation, and the bottom-resistance design loop has the most attenuation. For a maximum channel attenuation of 70 dB, useful channel bandwidths are 500 kHz, 1 MHz, and 1.5 MHz for resistance design, CSA, and mid-CSA loops, respectively.

Figure 13.45 Transfer Function of Resistance Design, CSA, and Mid-CSA Loops



13.3.1 ADSL Noise Environments

Noise environments were proposed for simulation study and testing of resistance design, CSA, and mid-CSA loops. The parameters for the loops were as follows:

- *Resistance design loops.* 24 ISDN Basic Rate Access NEXT disturbers, 24 ADSL FEXT disturbers, and –140 dBm/Hz of white background noise (see Figure 13.46)
- *CSA loops.* 10 HDSL NEXT disturbers, 10 ADSL FEXT disturbers, and –140 dBm/Hz of white background noise (see Figure 13.47)
- *Mid-CSA loops.* 24 ISDN Basic Rate Access NEXT disturbers, 24 T1 NEXT disturbers in an adjacent binder group, and –140 dBm/Hz of white background noise (see Figure 13.48)

Figure 13.46 Noise Environment for Resistance Design Loops

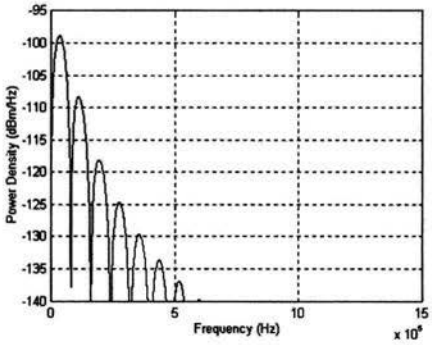


Figure 13.47 Noise Environment for CSA Loops

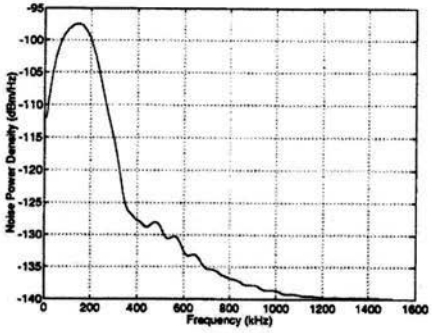
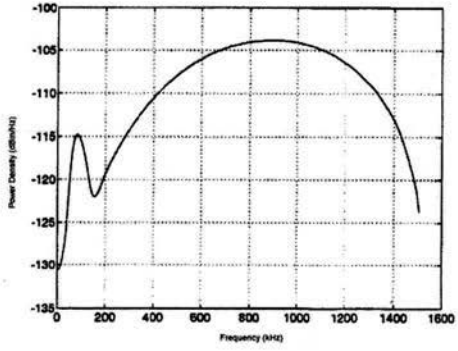


Figure 13.48 Noise Environment for Mid-CSA Loops



13.3.2 DMT ADSL Transmission Throughput Estimation

The transmission throughput of a DMT ADSL system can be estimated based on measured SNR of each subcarrier. The total transmission throughput, R, is the product of the prefix-modified baud rate and the number of bits each baud carries, as shown in the following expression:

$$R = \frac{1}{T}\sum_{i=1}^{256}\log_2\left(1+\frac{SNR_j}{9.55}\right)$$

The values are as follows:

$\Gamma = 9.55$ is chosen corresponding to an error rate of 10^{-7} without any coding gain.

SNR_j is the subcarrier-signal-to-noise-power ratio.

$1/T$ is the modified baud rate.

We have

$$\frac{1}{T} = \frac{2.2\times10^6}{512+32} = 4044Hz$$

A flat transmit power spectrum density of –40 dBm/Hz is also assumed. The SNR_j is calculated according to transmit power density, loop attenuation, and noise power density. Figures 13.49 and 13.50 show performance margins of downstream ADSL with a transmission throughput of 1.544 Mbps for 15 resistance design DSL test loops with and without echo cancellation, respectively. The average performance difference between with and without echo cancellation is about 5 dB.

Figures 13.51 and 13.52 show performance margins of downstream ADSL with transmission throughput of 6.4 Mbps (DS2+C+BRA), 6.6 Mbps (DS2+C+H0), and 6.8 Mbps (DS2+C+H0+BRA) for eight CSA test loops with and without echo cancellation. The average performance difference between with and without echo cancellation is also about 5 dB.

Figure 13.49 ADSL Performance Margin (with Echo Cancellation) for Resistance Design Loops

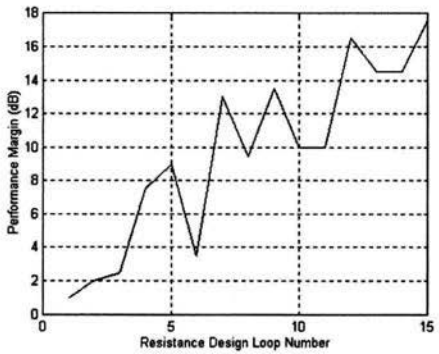
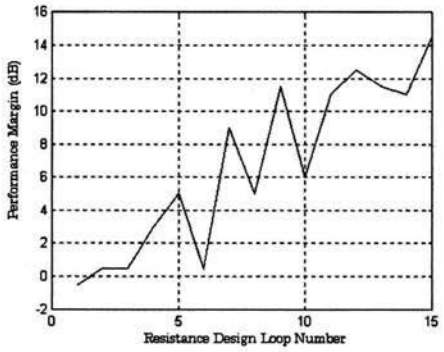


Figure 13.50 ADSL Performance Margin (without Echo Cancellation) for Resistance Design Loops



Figures 13.53 and 13.54 show performance margins of downstream ADSL with transmission throughput of 6.4 Mbps (DS2+C+BRA), 6.6 Mbps (DS2+C+H0), and 6.8 Mbps (DS2+C+H0+BRA) for six mid-CSA test loops with and without echo cancellation, respectively. The average performance difference between with and without echo cancellation again is also about 5 dB.

Figure 13.51 ADSL Performance Margin (with Echo Cancellation) for CSA Loops

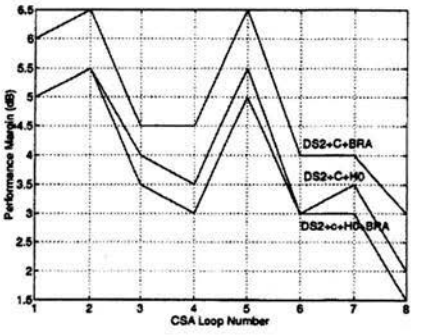


Figure 13.52 ADSL Performance Margin (without Echo Cancellation) for CSA Loops

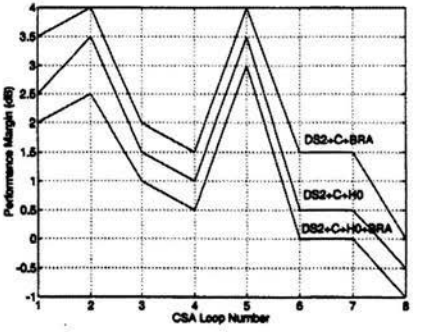


Figure 13.53 ADSL Performance Margin (with Echo Cancellation) for Mid-CSA Loops

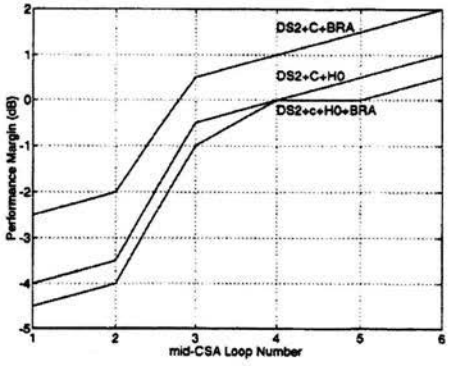
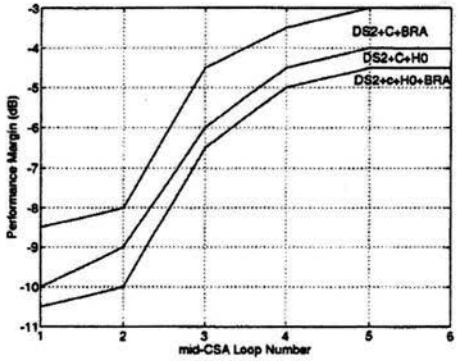


Figure 13.54 ADSL Performance Margin (without Echo Cancellation) for Mid-CSA Loops



13.3.3 *Estimating CAP ADSL Performance*

The transmission throughput of a CAP ADSL depends on the combination of the baud rate and the constellation size. Assuming that the fractional level signaling is possible,[109] the transmission throughput can be optimized by selecting a particular baud rate and corresponding constellation size based on a target error rate performance requirement.[110] Similarly, the performance margin of CAP ADSL can also be optimized by selecting a particular baud rate and corresponding constellation size based on a target transmission throughput with a defined error rate.

The performance of a CAP ADSL can be estimated based on MMSE of a DFE assuming equalizers have sufficient number of filter coefficients. The MMSE of a DFE under the combination of FEXT and white background noise is

$$\varepsilon_{\min} = \exp\left(\frac{1}{F_2 - F_1} \int_{F_1}^{F_2} \ln\left(1 + \frac{1}{8\times 10^{-20} l f^2 + \dfrac{N}{S^2 \left| H(f) \right|^2}} \right) df \right)$$

The values are as follows:

N is the background noise power.

l is the length of the loop.

$H(f)$ is the transfer function of the loop.

The integration bandwidth corresponds to the signaling baud rate— $\frac{1}{T} = F_2 - F_1$.

The corresponding SNR at the output of the DFE is

$$SNR = \frac{1-\varepsilon_{\min}}{\varepsilon_{\min}} \approx \frac{1}{\varepsilon_{\min}}$$

On the other hand, the required constellation size, M, is related to the baud rate, $\frac{1}{T}$, and the target transmission throughput, R_b, according to

$$M = 2^{R_b T}$$

The required $SNRr$ for that M value under a defined error rate is

$$SNR_r = \frac{\sigma^2}{3}(M-1) = \frac{\sigma^2}{3}\left(2^{R_b T} - 1\right)$$

The value σ^2 is related to the defined error rate. For a defined error rate of 10^{-7}, we can find the value of σ to be 5.333 according to the Gaussian distribution:

$$10^{-7} = \frac{2}{\sqrt{2\pi}} \int_{\sigma}^{\infty} e^{-\frac{x^2}{2}} dx$$

The performance margin is then the following:

$$SNR_m = SNR - SNR_r$$

13.4 Network and Premises Interfaces

ADSL was originally designed mainly for providing video-on-demand services. For this original application, an ATU-C accepts the 1.5 Mbps signal input and provides the low-speed control channel and the POTS channel. The POTS channel terminates on a conventional local switch. The 1.5 Mbps stream is provided by switched DS1 services. The low-speed control channel terminates on a packet handler, which in turn communicates with the DS1 switch and the Information Service Provider, as depicted in Figure 13.55.[111]

Figure 13.55 A Switch-Based Central Office ADSL Interface

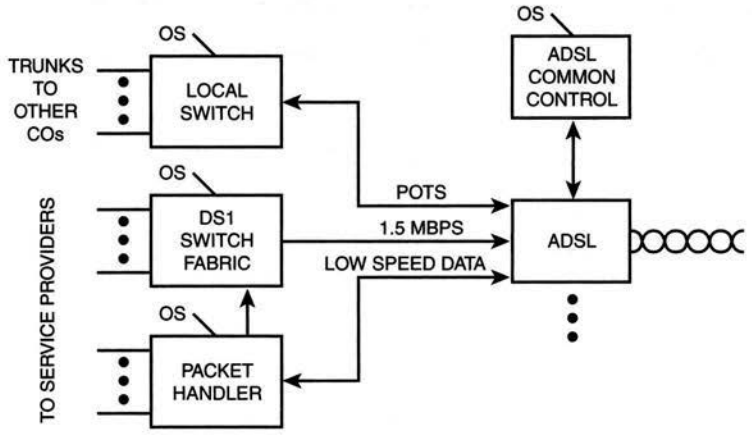
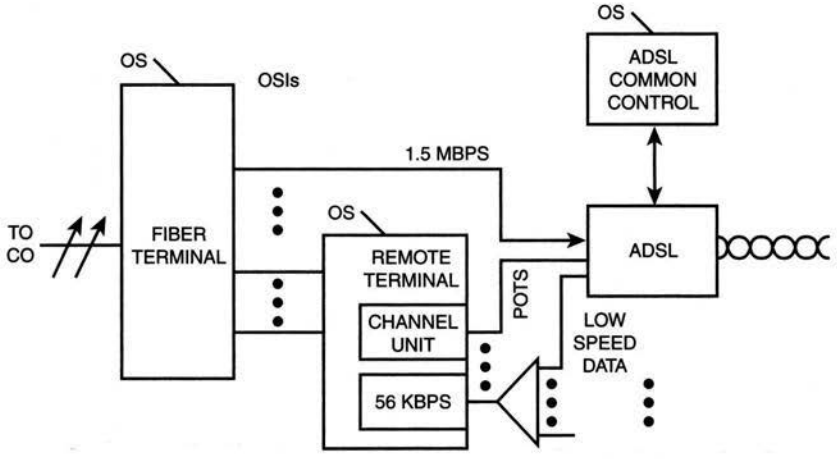


Figure 13.56 shows how the ADSL can be used with remote electronics.

Figure 13.56 A Switch-Based Remote Terminal ADSL Interface



The POTS service terminates on a conventional plug-in in a digital loop carrier remote terminal. The 1.5 Mbps signal is carried over a spare DS1 channel on the fiber multiplexer that feeds the remote electronics site. The low-speed data channel terminates on a data channel unit in the remote terminal or perhaps a number of ADSL low-speed control channels are concentrated first before being transported over a 56 kbps data channel.

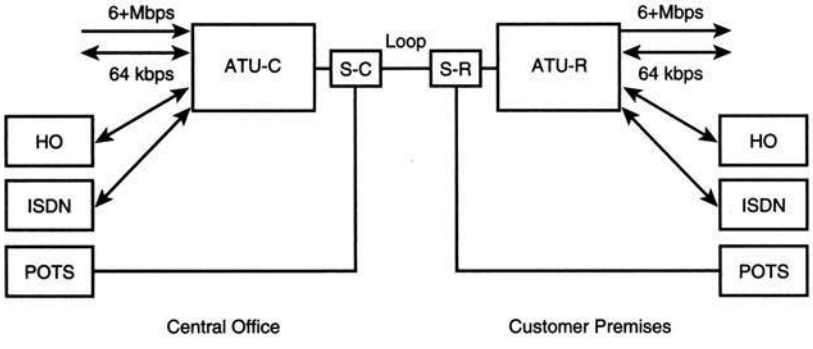
To enable the use of existing data protocols, it is desirable for the low speed control channel to be full duplex. At least two data formats were considered. The control channel could be implemented with an X.25 protocol, which could run at relatively low speeds, such as 9.6 kbps. The network transport is widely available for the X.25 protocol. The control channel could also be implemented through an ISDN Basic Rate D channel. The ISDN Basic Rate services are available from most Central Offices.

Figures 13.55 and 13.56 include an ADSL common controller that communicates with a number of ADSL office units. The office units in turn communicate with the remote units by way of an overhead channel. This overhead facilitates system synchronization and maintenance. Transmission performance and alarm conditions can be collected and forwarded to an operations system (OS). Eventually, many of the functions shown in Figures 13.55 and 13.56 can be integrated into one system, in which the ADSL office units become "line cards" in an advanced switch or remote terminal.

A DMT ATU-C can have an interface with up to seven different digital channels in addition to the POTS channel. This interface includes up to four asymmetrical channels, one control channel of 16 kbps or 64 kbps, one ISDN channel with 2B+D subchannels, and one H0 telephony channel that can be further divided into six B channels. Figure 13.57 depicts interfaces for a DMT ATU-C and an ATU-R. Most of these different digital channels can be handled by DS1 and ISDN interfaces for the video-on-demand and telephony services.

The low speed control channel can be either handled by the X.25 protocol or treated as a D channel. The POTS channel needs to be connected through the conventional switch. In short, the DMT ATU-C interface can be handled in a way similar to that of the 1.5 Mbps ADSL, but with a higher digital transmission throughput bandwidth.

Figure 13.57 DMT ATU-C and ATU-R Interfaces



13.4.1 Central Office Service Capabilities

A Central Office number (the middle three digits of YYY as in a telephone number of (XXX) YYY-ZZZ) can have an average of about 5,000 subscribers. A Central Office with multiple CO numbers could have more subscribers. A Distribution Area (DA) can serve 200–600 subscribers. A Remote Terminal (RT) with Optical to Electrical (O/E) and Electrical to Optical (E/O) conversion and power supply equipment, can serve four to six DAs. In other words, an RT might need to serve about 3,000 subscribers; however, it is not likely that every subscriber will have ADSL service, especially at the early introductory stage. We call the ADSL service offering among all subscribers the *take rate* that could range from 5 to 25 percent initially.

Compared with the digitized voice service at 64 kbps, the throughput of a 1.72 Mbps ADSL, including overhead, is 26.875 times higher. The throughput of a 6.784 Mbps ADSL, also including overhead, is 106 times higher than digitized voice service. Considering a take rate of 10 percent, the required total throughput at a CO in the downstream direction will be 860 Mbps (5000×0.1×1.72) and 3.392 Gbps (5000×0.1×6.784). At an RT, the downstream will be 510 Mbps (3000×0.1×1.72) and 2.0352 Gbps (3000×0.1×6.784) for 1.72 Mbps and 6.784 Mbps ADSL systems, respectively. These calculations for the CO and RT represent a throughput upgrade of 2.6875 and 10.6 times, respectively.

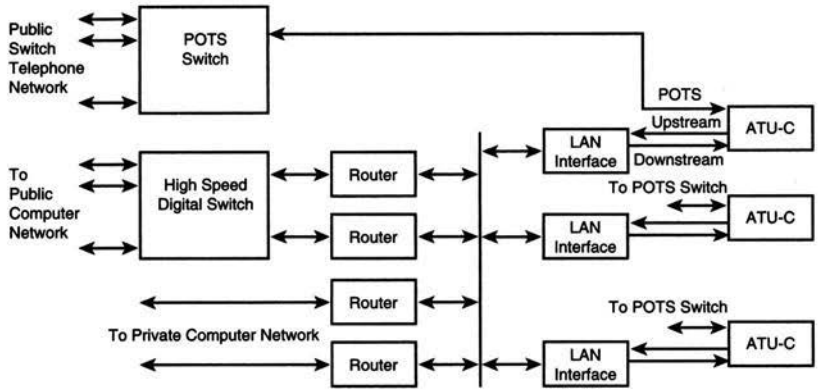
On the other hand, the throughput upgrade can be much lower than the previous calculation for a CO and RT if only additional data traffic is considered, depending on delay and packet loss requirements. As a reference, the optical equipment that can be used to connect between multiple COs and between a CO and RTs have throughputs of 155.52 Mbps, OC-3, and 622.08 Mbps, OC-12, and so forth. Hence, depending on the RT size and the take rate, a number of OC-3 or OC-12 links is more than likely.

For ATM or TCP/IP-based data traffic, the dedicated digital interface might not be efficient and necessary. Therefore, all ADSL channels can be first statistically multiplexed through a Local Area Network (LAN) at the CO (see Figure 13.58) before being connected to the backbone digital network. A LAN interface is required for each ADSL transceiver. The traffic external to the LAN is connected to a public or private computer network through a router. The concentrated data traffic might or might not be connected to the computer network through a digital switch.

Because all ATU-Cs are close to each other and the LAN is only served as a traffic concentrator, the function of LAN and router can be combined to form a special-purpose ADSL ATU-C concentrator. This concentrator might evolve to a next-generation high-throughput

digital switch if the ATM protocol can be implemented over the concentrator and over ADSL links.

Figure 13.58 Router and LAN Computer Connection



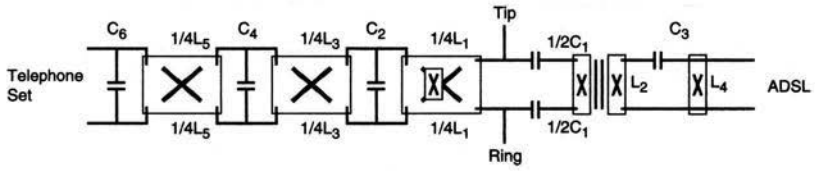
13.4.2 *POTS Splitters*

A POTS splitter separates the 300 Hz to 3,500 Hz voice channel from the upstream and downstream channels. In the voice passband, the POTS splitter should pass not only voice frequency signal with little distortion, but also dial tone, ringing, and on/off hook signals. For the voice frequency signal, the POTS splitter should maintain the original POTS impedance and return loss requirements. Otherwise, POTS automatic network test systems would have difficulty troubleshooting telephone subscriber loop and residential equipment problems. The POTS splitter interface also needs to pass high voltage and high current signals for ringing, on/off hook, and so forth.

The structure of a POTS splitter is similar for both ATU-C and ATU-R. A POTS splitter consists of mainly a lowpass filter for the POTS interface and a highpass filter for upstream and downstream channels. The lowpass filter removes the interference from upstream and downstream channels to the POTS channel. The POTS channel impedance, which is about 600 ohms, should not be affected by the impedance of upstream and downstream channels, which is about 100 ohms. The highpass filter removes the interference from the POTS channel to upstream and downstream channels. The harmonics of ringing and on/off hook signals should especially be prevented from entering upstream and downstream channels.

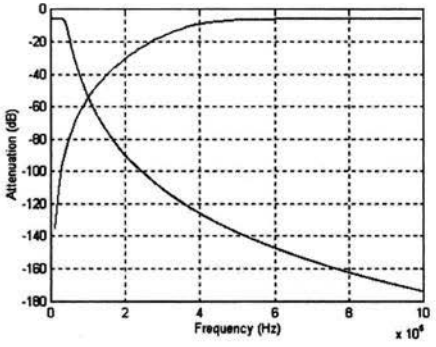
Figure 13.59 shows the circuit diagram of a POTS splitter.[112]

Figure 13.59 Circuit Diagram of a POTS Splitter



The inductors for the lowpass filter should have high Q for small signal attenuation and high current saturation levels to avoid generating harmonics under the heavy drive of high-intensity voltage or current. A stopband loss of 70 dB is required for both lowpass and highpass filters. Because of the narrow guard band, a slop of –60 dB per decade is also required. Figure 13.60 shows required lowpass and highpass frequency responses for a POTS splitter.

Figure 13.60 Desired Lowpass and Highpass Frequency Responses for a POTS Splitter



The highpass filter for ADSL is at the right side of the POTS input, and the lowpass filter for the telephone set is at the left side, as illustrated in Figure 13.59. Only POTS and telephone interfaces need balanced design. The highpass filter is a fourth-order filter with balanced input for the POTS line and unbalanced output for the ADSL transceiver. The lowpass filter is a seventh-order filter with balanced input and output ports. These filters can be first designed according to the conventional unbalanced structure.

The capacitance for the balanced part of the highpass filter is simply twice that calculated for the unbalanced structure. The inductance of the lowpass filter is a quarter of that calculated for the unbalanced structure, assuming a coupling coefficient of approximately one between two inductors sharing the same magnetic core.

The lowpass filter part of the POTS splitter is an LC ladder. Values for Ls and Cs can be calculated according to the following formula:

$$L_{2m-1} = \frac{2R_1 \sin\left(\frac{(4m-3)\pi}{2n}\right)}{\omega_c}$$

$$C_{2m} = \frac{2\sin\left(\frac{(4m-1)\pi}{2n}\right)}{R_1\omega_c}$$

For

m=1,2,3

n=6

R_1=600

ωc =2 π4 kHz

we have

L_1=2.06 mH

C_2=0.5627 μF

L_3=7.68 mH

C_4=0.768 μF

L_5=5.627 mH

C_6=0.206 μF

These L and C values result to a Butterworth filter frequency response. A fourth-order lowpass filter can be first designed and then transformed to a highpass filter.

For

m=1,2

n=4

R_1=100

ωc =2 π 40 kHz

we have

$$L'_1 = 0.3045 \text{ mH}$$
$$C'_2 = 0.07352\ \mu\text{F}$$
$$L'_3 = 0.7352 \text{ mH}$$
$$C'_4 = 0.03045\ \mu\text{F}$$

The corresponding highpass filter component values are as follows:

$$C_1 = \frac{1}{\omega_c^3 L'_1} = 0.052 \text{ mF}$$
$$L_2 = \frac{1}{\omega_c^3 C'_2} = 0.21 \text{ mH}$$
$$C_3 = \frac{1}{\omega_c^3 L'_2} = 0.0215 \text{ mF}$$
$$L_4 = \frac{1}{\omega_c^3 C'_4} = 0.52 \text{ mH}$$

Values for Ls and Cs can also be calculated according to the following formula:

$$L_1 = \frac{2R_1 \sin\left(\frac{\pi}{2n}\right)}{(\sinh a - \sinh \tilde{a})\omega_c}$$
$$L_{2m-1}C_{2m} = \frac{4 \sin \gamma_{4m-3} \sin \gamma_{4m-1}}{\omega_c^2 f_{2m-1}(\sinh a, \sinh \tilde{a})}$$
$$L_{2m+1}C_{2m} = \frac{4 \sin \gamma_{4m-1} \sin \gamma_{4m+1}}{\omega_c^2 f_{2m}(\sinh a, \sinh \tilde{a})}$$

in which

$$\gamma_m = \frac{m\pi}{2n}$$
$$\tilde{a} = \frac{1}{n}\sinh^{-1}\left(\frac{\sqrt{1-K_n}}{\varepsilon}\right)$$

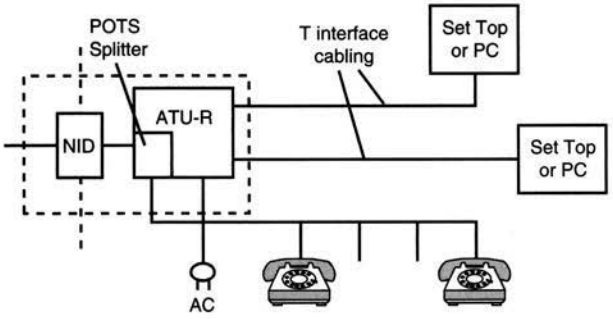
$$f_m(u,v) = u^2 + v^2 + \sin^2 \gamma_{2m} - 2uv\cos\gamma_{2m}$$

for a Chebyshev network.

13.4.3 *Application of POTS Splitter*

A POTS splitter could be included in an ATU-R, as shown in Figure 13.61.

Figure 13.61 An ATU-R with a Built-in POTS Splitter

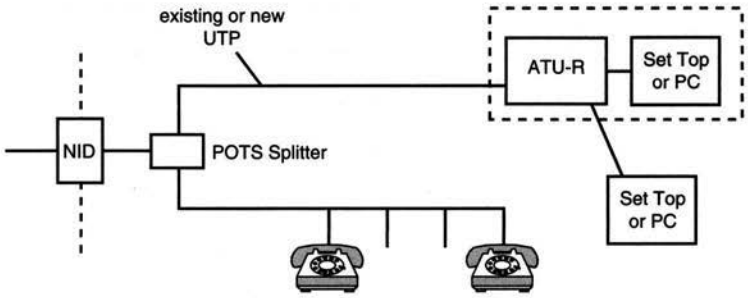


The ATU-R interfaces to the single loop pair on one side and demultiplexes POTS, control, and asymmetrical channels on the other side. The POTS channel is routed to the customer telephones, answering or facsimile machines, analog modems, and so forth, in the conventional fashion. Each asymmetrical channel should be connected to a service module.

The nature of the service module will vary from application to application. For example, the service module might include an MPEG video decoder, fast packet hardware, or CD-audio hardware. The output of the service module might be a baseband video signal, a packet protocol, or an audio signal. The low-speed control channel must also be connected to all service modules. In computer-based applications, the service module functionality may reside on a card that plugs into a PC.

In Figure 13.61, NID stands for *Network Interface Device,* which is available for conventional POTS lines and is located at the phone wire entry point. Similar to the case of ISDN, ADSL T interfaces can be defined to connect an ATU-R to service modules. A POTS splitter also can be implemented external to an ATU-R (see Figure 13.62). This configuration provides the flexibility for the location of ATU-R, while avoiding the direct transmission of an ADSL signal over low-quality telephone in-house wiring. Low-quality in-house wiring with open bridge taps can easily damage the performance of an ADSL system.

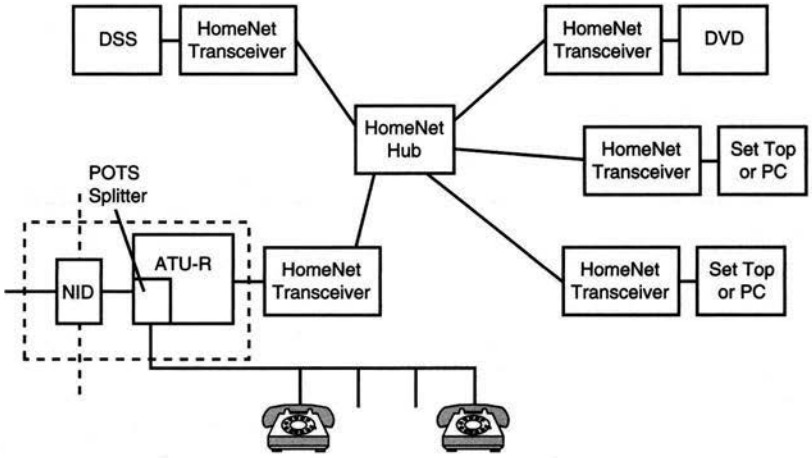
Figure 13.62 An ATU-R with a Separate POTS Splitter



The definition of the ADSL T interface is not an easy task. While complexity is required for the acceptance of a consumer-oriented market, the ADSL T interface needs a combined transmission throughput of more than 7 Mbps with bi-directional multiple access points. For computer applications, the function of the T interface can be satisfied with a LAN, such as 10BaseT or 100BaseTX. To carry video or telephony—*isochronous traffic*—the concept of a high-throughput digital home network can be adapted to meet the demand of the ADSL T interface.

The high-throughput home network can be implemented with star topology data grade twisted-pair cable based on the transmission protocol of IEEE P1394.[113] The home network can also be shared with other network or local information resources. With a home network, a consumer does not have to struggle with deciding which terminal to be connected to which access network, as shown in Figure 13.63.

Figure 13.63 ADSL Connected to a Home Network



13.5 ADSL Standards (T1.413)

The main purpose of the T1.413 document is to describe interfaces of the DMT ADSL modem at the Central Office (CO) side and at the subscriber side. The extended interface covers not only pins and voltage levels, but also frame structure, coding, interleaving, channelization as well as initialization, bit loading, and online adaptation.

13.5.1 Frame Structure

The standard DMT ADSL was originally designed mainly for the video-on-demand application. It was observed that the minimum throughput for a compressed video channel is 1.544 Mbps (MPEG-I), and higher throughput (3 ~ 6 Mbps MPEG-II) might also be necessary to carry live programs. Bearing its name, ADSL mainly carries asymmetrical video-on-demand signals. For the video-on-demand application, the downstream throughput (from a CO to a subscriber) needs to be high (1.544 ~ 6.176 Mbps). Conversely, the upstream throughput requirement is to satisfy the need of control signals (16 ~ 64 kbps). For the need of high-throughput telephony applications, a symmetrical ISDN Basic Rate Access (128 kbps) channel and a symmetrical H0 (384 kbps) channel are also included.

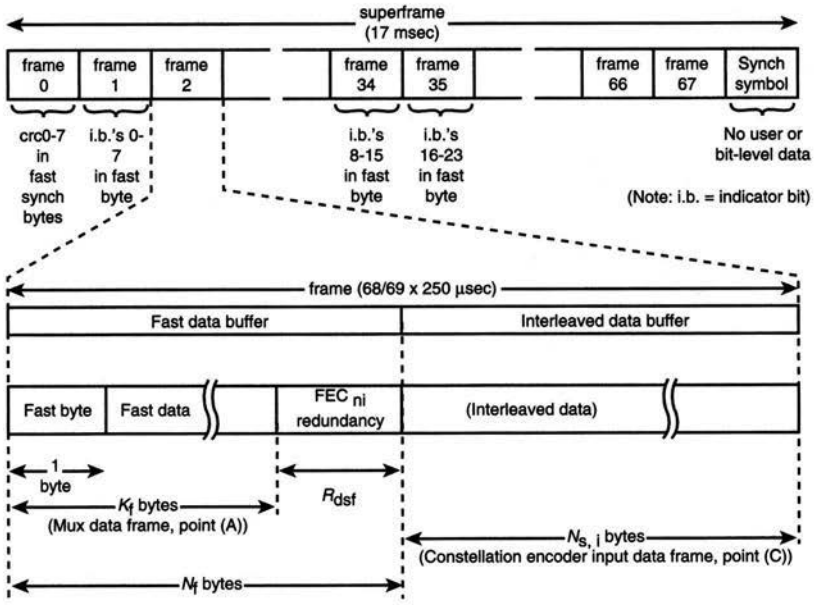
To distinguish these channels for potential applications, a flexible frame structure with different options was defined. Due to the IFFT/FFT operation, the frame is only defined in retrieval sequence index for input/output interfaces, but has no real-time index as other systems do. For the downstream direction, a super frame is defined for every 68 IFFT/FFT operations. A synchronization frame is also created for every super frame. Therefore, there are 69 frames for every super frame. Because of the extra synchronization frame, in real time, every frame is not exactly aligned with every IFFT/FFT operation. In other words, a frame is not aligned with each IFFT/FFT operation.

In the downstream direction and for a baud rate of 4 kHz, the super frame has a time duration of 68/4000=17 milliseconds. Due to the added synchronization frame, the frame rate on the line becomes 69/68×4000. The sampling rate would have been 512×4000 Hz without cyclic prefix. With the addition of 40 prefix sampling points, the sampling rate is (512+40)×4000=2,208,000 Hz.

In practice, the prefix length is 32 and the reduction of eight sampling points every frame for 68 frames generates a synchronization frame of size 8×68=544 sampling points, which is the same size as a data frame with cyclic prefix. Therefore, we can also represent the sampling rate as (512+32)×69/68×4000=2,208,000 Hz.

Figure 13.64 shows the general structure of a frame as well as its relationship to the super frame.

Figure 13.64 DMT ADSL Frame Structure

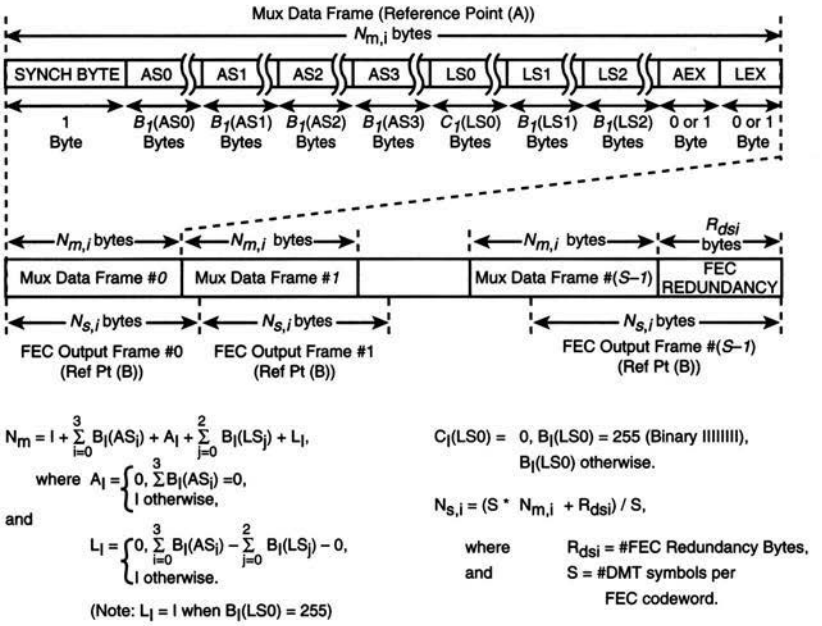


As shown in Figure 13.64, each frame consists of a fast data portion, for delay-sensitive applications, and an interleaved data portion, for video-on-demand applications. Depending on the transmission throughput, the number of data bytes for each frame can be different. On the other hand, the number of total bytes for each frame other than the synchronization frame is kept the same by inserting dummy bytes. The first byte of every frame is used for fast data, which could be for CRC bits, indication bits for OAM functions, *embedded operation channel* (eoc) bits, or synchronization control bits for fast data.

There are eight CRC bits, which are always located at the first byte of the first frame (frame 0), and 24 indicator bits, which are located at first bytes of frame 1, 34, and 35. The remaining first bytes of frames 2 to 33 and 36 to 67 are used for either eoc (embedded operation channel) or synchronization control. An eoc byte is indicated by a value 1 of bit 0. A synchronization control byte is indicated by a value 0 of bit 0. Both eoc and synchronization control bytes appear in a byte pair—an even byte followed by an odd byte.

Figure 13.65 indicates that FEC for interleaved data is appended at the end of a group of frames, and therefore the interleaved data logic frame where the channelization is defined does not share the frame boundary with that of IFFT/FFT frame. The size of the group for FEC is decided by the interleaving depth.

Figure 13.65 Frame Structure of Downstream Interleaved Data



$$N_m = I + \sum_{i=0}^{3} B_I(AS_i) + A_I + \sum_{j=0}^{2} B_I(LS_j) + L_I,$$

where $A_I = \begin{cases} 0, \sum B_I(AS_i) = 0, \\ I \text{ otherwise,} \end{cases}$

and

$$L_I = \begin{cases} 0, \sum_{i=0}^{3} B_I(AS_i) - \sum_{j=0}^{2} B_I(LS_j) - 0, \\ I \text{ otherwise.} \end{cases}$$

(Note: $L_I = I$ when $B_I(LS0) = 255$)

$C_I(LS0) =$ 0, $B_I(LS0) = 255$ (Binary IIIIIIII),
$B_I(LS0)$ otherwise.

$N_{s,i} = (S * N_{m,i} + R_{dsi}) / S,$

where R_{dsi} = #FEC Redundancy Bytes,
and S = #DMT symbols per FEC codeword.

For the interleaved data, the first byte is the SYNCH byte. The SYNCH byte of the first frame, frame 0, is for eight bits of CRC. The remaining SYNCH bytes are used either for synch control or aoc (ADSL overhead control). The synch control byte is valid as long as the interleave portion is occupied with data; otherwise, the aoc is presented. Both fast data and interleaved data portions can accommodate four asymmetrical and three symmetrical channels. At the end of fast data and interleaved data portions, there are stuffing AEX and LEX bytes for the synchronization control of each asymmetrical and symmetrical channel, respectively.

The channels, from AX0 through LX2, operate at rates that are multiples of the symbol rate. Therefore, the throughput of each channel might not be exactly the same as that of the backbone network. The AEX and LEX bytes regulate the minor throughput differences using the *bit stuffing technique*. That is, if the throughput of the channel is higher than that of the network, a dummy byte will be inserted/removed in the data stream. The ADSL standard also enables the deletion of a byte from the channel. The addition will match to a lower network throughput, while the deletion will match to a faster network throughput, but with the danger of losing data bytes.

For fast data buffers, the indication of which channel and the corresponding operation of bit insertion/deletion information is in the fast byte. For interleaved data buffer, the same information is in the synchronization byte. Both the fast byte and the synchronization byte offer the option of being used as an eoc frame and a ADSL overhead control channel, respectively.

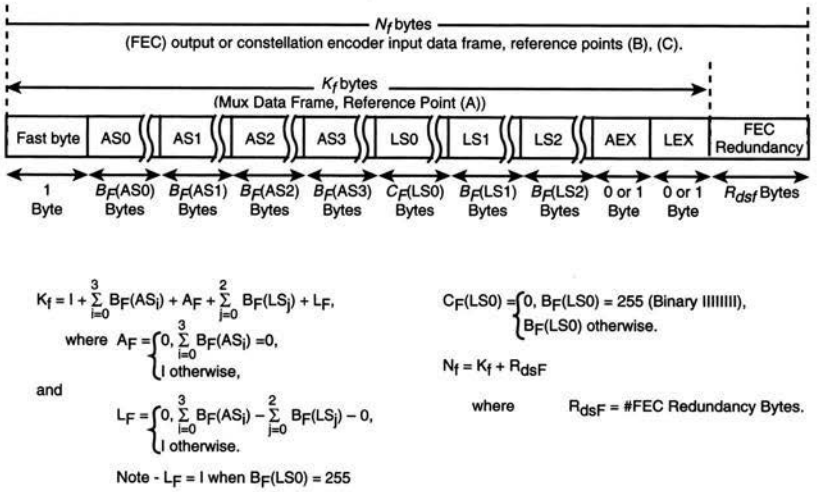
ADSL can have up to four simplex channels (AS0, AS1, AS2, and AS3) that are causes of asymmetrical throughput, and three duplex channels (control channels LS0, LS1, and LS2). Depending on the class of operation—related to the transmission environment and the transceiver capacity—some of these channels can be activated in the initialization process with possibly a few different optional transmission throughputs. For an interface to the North America T digital hierarchy system, there could be four classes of channel and throughput combinations, as explained in the following list.

- *Class 1.* The maximum simplex throughput is 6.144 Mbps for up to four channels (AS0, AS1, AS2, AS3); the maximum duplex channel throughput is 640 kbps for up to two channels (LS1, LS2); and the control channel (LS0) throughput is 64 kbps.
- *Class 2.* The maximum simplex throughput is 4.608 Mbps for up to three channels (AS0, AS1, AS2); the maximum duplex channel throughput is 608 kbps for one channel (LS1 or LS2); and the control channel (LS0) throughput is 64 kbps.
- *Class 3.* The maximum simplex throughput is 3.072 Mbps for up to two channels (AS0, AS1); the maximum duplex channel throughput is 608 kbps for one channel (LS1 or LS2); and the control channel (LS0) throughput is also 64 kbps.
- *Class 4.* The simplex throughput is 1.536 Mbps for one channel (AS0); the duplex channel throughput is 160 kbps for one channel (LS1); and the control channel throughput is 16 kbps, which is not implemented in the control channel slot but through the synchronization overhead.

Multiplexing slots for both simplex and duplex channels are available in both fast and interleaved parts of every frame in a similar order, that is, starting with AS0 and followed by AS1, AS2, AS3, LS0, LS1, and LS2. The fast data buffer starts with a fast byte and ends with a few FEC redundancy bytes, as illustrated by Figure 13.66.

There also could be an AEX and/or LEX byte before FEC redundancy bytes. The interleaved data buffer starts with a synchronization byte and could end with AEX and/or LEX bytes. A group of FEC redundancy bytes for the interleaved buffer is attached at the end of the interleave frame, which has a number of frames equal to the interleaving depth.

Figure 13.66 Channelization of Downstream Fast Data



For class 1 transport, by default, both AS0 and AS1 channels should have 96 bytes each, the LS0 channel should have two bytes in the interleaved data buffer, the LS1 channel should have five bytes, and the LS2 channel should have 12 bytes in the fast data buffer.

For class 2 transport, by default, the AS0 channel should have 96 bytes, the AS1 channel should have 48 bytes, the LS0 channel should have two bytes in the interleaved data buffer, and the LS2 channel should have 12 bytes in the fast data buffer.

For class 3 transport, by default, both AS0 and AS1 channels should have 48 bytes each, the LS0 channel should have two bytes in the interleaved data buffer, and the LS2 channel should have 12 bytes in the fast data buffer.

For class 4 transport, by default, the AS0 channel should have 48 bytes in the interleaved data buffer and the LS1 channel should have five bytes in the fast data buffer.

A pair of bytes, B_I and B_F, is used for each channel, AS0 through LS2, in the third phase of the initialization process, which is discussed shortly. This byte pair indicates the number of bytes in either interleaved or fast data buffer. The allocation of each channel is exclusive, meaning that a channel can be either in fast data buffer or interleaved data buffer, but not in both. For the class 1 transport, the 16 kbps control channel can be either in interleaved or fast data buffer by designating either B_I=255 and B_F=0 or B_I=0 and B_F=255. For the default class 1 transport, we have the following byte pairs:

- B_I(AX0)=60 and B_F(AX0)=00

- B_I(AX1)=60 and B_F(AX1)=00
- B_I(AX2)=00 and B_F(AX2)=00
- B_I(AX3)=00 and B_F(AX3)=00
- B_I(LX0)=02 and B_F(LX0)=00
- B_I(LX1)=00 and B_F(LX1)=05
- B_I(LX2)=00 and B_F(LX0)=0C

In the upstream direction, there are no simplex channels and no AEX bytes. There could be a maximum of three channels and, therefore, three pairs of B_F and B_I bytes. Otherwise, the upstream super frame structure is similar to that of downstream, as illustrated in Figure 13.67.

Figure 13.67 Frame Structure of Upstream Interleaved Data

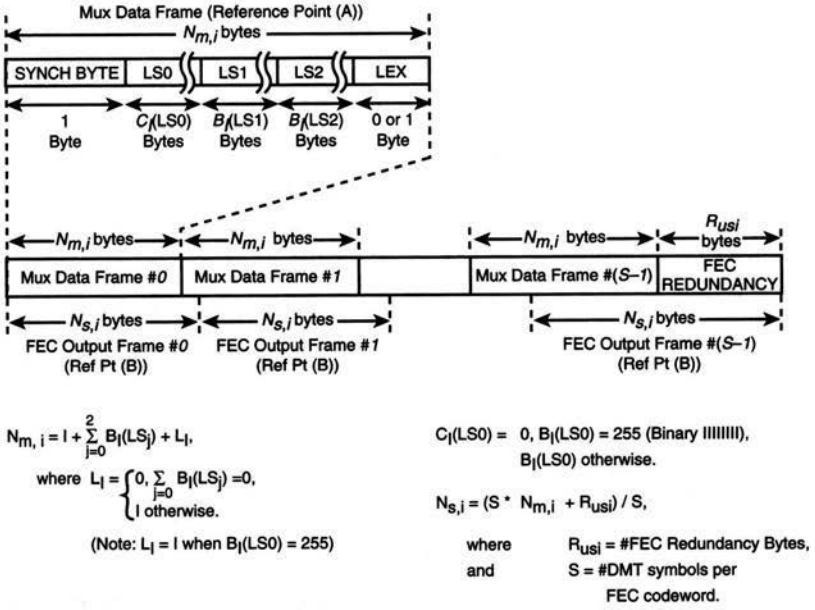
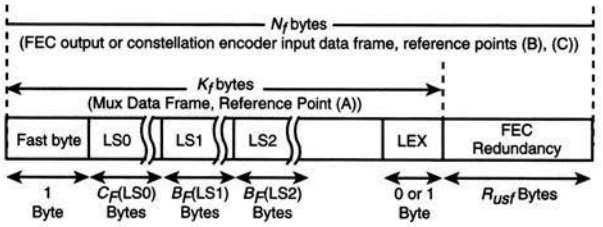


Figure 13.68 shows the channelization of upstream fast data. Similar to the downstream, the upstream fast data buffer starts with a fast byte and ends with a few FEC redundancy bytes.

Figure 13.68 Channelization of Upstream Fast Data



$$K_f = l + \sum_{j=0}^{2} B_F(LS_j) + L_F,$$

where $L_F = \begin{cases} 0, & \sum_{j=0}^{2} B_F(LS_j) = 0, \\ l & \text{otherwise.} \end{cases}$

Note - $L_F = l$ when $B_F(LS0) = 255$

$$C_F(LS0) = \begin{cases} 0, & B_F(LS0) = 255, \\ B_F(LS0) & \text{otherwise.} \end{cases}$$

$$N_f = K_f + R_{usf},$$

where R_{usf} = #FEC Redundancy Bytes.

Although the upstream data baud rate is the same as that of the downstream data baud rate, the IFFT/FFT size is 64 and the sampling rate is correspondingly lower. In addition, due to the added synchronization frame, the frame rate on the line is 69/68×4000. The sampling rate would have been 64×4000 Hz without cyclic prefix. With the addition of five prefix sampling points, the sampling rate is (64+5)×4000=276,000 Hz. In practice, the prefix length is four and the reduction of one sampling point every frame for 68 frames generates a synchronization frame of size 68 sampling points, which is the same size as a data frame with cyclic prefix. Therefore, we can also represent the sampling rate as (64+4)×69/68×4000=276,000 Hz.

13.5.2 Interleaving

ADSL interleaving depths are 16, 32, and 64 FEC codewords for transport class 4, class 3, and classes 2 and 1, respectively. We also have the FEC message word size of 208, 212, 158, and 210 for transport class 4, class 3, class 2, and class 1, respectively. Therefore, the interleaving depths can also be described as follows:

- Class 4: 16×208=3,328 bytes
- Class 3: 32×212=6,784 bytes
- Class 2: 64×158=10,112 bytes
- Class 1: 64×210=13,448 bytes.

The interleaving can be implemented in a circular shift register or a dual port memory. For the shift register implementation, the first byte of every codeword should stay where it is and the rest bytes should be delayed by $(i-1)\times(N-1)$ bytes, in which i is the position of each byte, and N is the interleaving depth. For an interleaving depth of 64, the first byte of every codeword should have no delay, the second byte should be delayed by 63 bytes, the third byte should be delayed by 126 bytes, and so on.

13.5.3 Reed-Solomon Coding

The forward error correction (FEC) coding processes, using Reed-Solomon code, can be applied for both fast data and interleaved data. For fast data, there are four FEC redundancy bytes corresponding to 5 or 17 fast data bytes. For interleaved data, there could be 16 FEC redundancy bytes for 194 interleaved data bytes, 12 FEC redundancy bytes for 146 interleaved data bytes, 16 FEC redundancy bytes for 196 interleaved data bytes, or 16 FEC redundancy bytes for 192 interleaved data bytes. The code is based on the GF(256) Galois field arithmetic.

13.5.4 Bit Encoding

Each subchannel is designed for a maximum number of 15 bits corresponding to a 32,768 QAM constellation whenever the subchannel SNR permits. For an even number of bits, the constellation is square. Constellations are defined for n=2, 3 bits and a recursive procedure is also designed for n>3. The optional Wei's 16-state four-dimensional Trellis code can be used to improve the transmission performance.

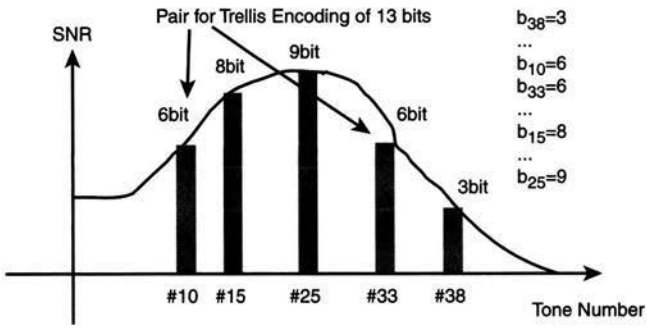
Clipping, due to the limited precision of ADC and DAC, is most likely to cause errors on those tones that, in anticipation of higher received SNRs, have been assigned the largest number of bits (and therefore have the densest constellations). These occasional errors can be reliably corrected by the FEC coding if the tones with the largest number of bits have been assigned to the interleave buffer.

The *tone-ordered* encoding assigns the first B_F bytes from the symbol buffer to the tones with the smallest number of bits assigned to them, and the remaining B_I bytes to the remaining tones. The number of bits allowed for each tone information is used to construct a bit/tone order table. We round all tones with 0 bit allowed at the beginning of the order table, with 1 bit allowed next, and so on.

Data bytes from the DMT symbol buffer are extracted according to the ordered bit/tone allocation table by least significant bit first. The Trellis coding is based on a pair of adjacent tones, according to the ordered bit/tone table. The Trellis coding process generates one additional bit per pair of tones. Because the bit/tone parameter is pre-calculated, one fewer

bit is extracted to make room for the extra coding bit. Figure 13.69 illustrates the tone ordering.

Figure 13.69 Tone Ordering



Bits extracted from the DMT symbol buffer are labeled as $u_1, u_2, \ldots u_{z-1}, u_z$, and are encoded into $w_0, w_1, \ldots w_{y-1}, v_0, v_1, \ldots v_{z-y}$, for which z=x+y–1. x is the number of bits for one tone of the pair, and y is the number of bits for the other tone of the pair. We have x+y–1 bits coming into the encoder and x+y bits going out from the encoder. The relationships between u_0, u_1, u_2, u_3, and w_0, w_1, v_0, v_1, are defined as

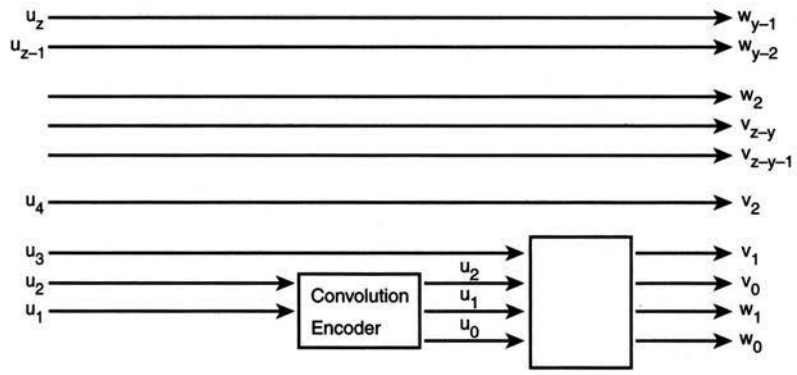
$$v_1 = u_1 \oplus u_3$$
$$v_0 = u_3$$
$$w_1 = u_0 \oplus u_1 \oplus u_2 \oplus u_3$$
$$w_0 = u_2 \oplus u_3$$

Figure 13.70 illustrates the DMT Trellis encoder.

For the Trellis coding, a minimum of two bits is used for each tone. For a pair of two tones, we have a minimum of four bits describing a four-dimensional space, which means there is only one position identified within that four-dimensional space by a group of four bits.

A two-dimensional Trellis encoder can be used to divide a two-dimensional space into four subsets. Within each subset, the space between each point is sufficient for the detection under a specific SNR. The function of the convolution encoder is such that the transition path of a signal sequence is made very distinctive from other possibilities due to possible misdetection of a particular signal point. By the same principle, ADSL uses a four-dimensional Trellis encoder to divide a four-dimensional space into eight subsets.

Figure 13.70 Trellis Encoding



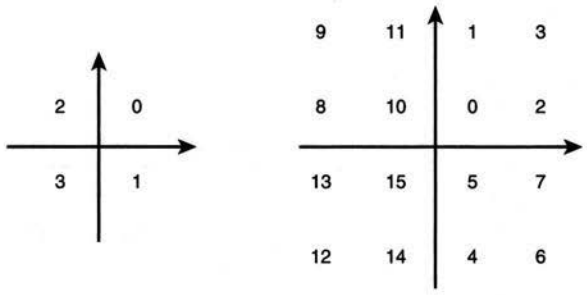
Each subcarrier can have its unique constellation depending on the SNR of that subcarrier. In general, a constellation with an even number of bits is square, and a constellation with an odd number of bits is almost square, but with corner points missing.

A minimum square constellation for even bits can be described with four numbers: 0, 1, 2, and 3. A square constellation with more than two bits can be generated recursively by replacing each label n by the 2×2 block of labels

$$\begin{matrix} 4n+1 & 4n+3 \\ 4n & 4n+2 \end{matrix}$$

Figure 13.71 illustrates an even bits constellation.

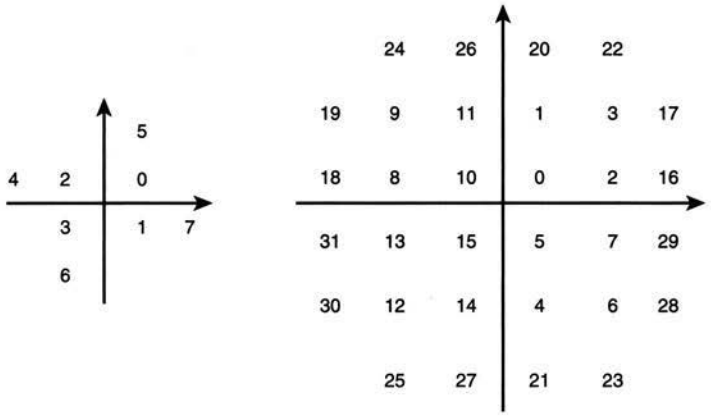
Figure 13.71 Even Bits Constellation



We can also describe a constellation by using b bits in digital or binary format. For $b=2$, we have 4QAM, for $b=4$, we have 16QAM, and so on. Coordinates of constellation points are $(v_{b-1}, v_{b-3},\ldots v_1, 1)$ and $(v_{b-2}, v_{b-4},\ldots v_0, 1)$ in binary format, for which v_{b-1} and v_{b-2} are sign bits with 1 for minus sign, and v_i can be either 0 or 1.

If b is odd and greater than 3, the 2 MSBs of X and the 2 MSBs of Y are determined by the 5 MSBs of b bits. Let c=(b+1)/2, the X and Y have the twos-complement binary representations $(X_c, X_{c-1}, v_{b-4}, v_{b-6},\ldots v_3, v_1, 1)$ and $(Y_c, Y_{c-1}, v_{b-5}, v_{b-7}, v_{b-9},\ldots v_2, v_0, 1)$ where Xc and Yc are the sign bits of X and Y, respectively. The same procedure can be used to construct the larger odd-bit constellation recursively. Figure 13.72 illustrates an odd bits constellation.

Figure 13.72 Odd Bits Constellation



Because of the cyclic prefix, the frequency spacing between adjacent subcarriers is 4.3125 kHz instead of 4 kHz. The cyclic prefix has 32 sample points compared with the IFFT size of 512. Carrier #64, at 276 kHz, is a pilot tone for timing recovery purposes. The tone is generated by sending (0,0) to the carrier #64 of a constant magnitude. The carrier #256 at the Nyquist frequency is not used for the data transmission purpose. The magnitude of subcarrier from #51 to #256 can be uniformly raised by either 3 or 6 dB to combat T1 Near End Crosstalk (NEXT). A fine gain adjustment of –1.5 to +1.5 dB can also be used to fit the SNR of each subcarrier while keeping the overall transmit power level constant.

13.5.5 *Scramblers*

There are separate scramblers for fast data and interleaved data. The separate scramblers use the same algorithm. The operation of the scrambler can be described as

$$d_n' = d \oplus d_{n-18}' \oplus d_{n-23}'$$

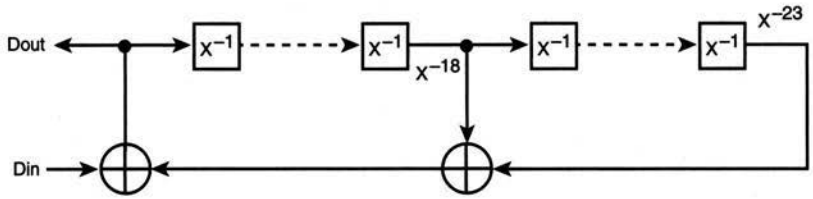
The values are as follows:

d_n is the nth output from the fast or interleaved buffer.

d_n' is the nth output from the corresponding scrambler.

Figure 13.73 shows the structure of the scrambler used by the DMT ADSL.

Figure 13.73 Scrambler Structure



These scramblers shall be applied to the serial data stream without reference to any framing or symbol synchronization. The same expression is applied for both downstream and upstream directions.

13.5.6 Initialization

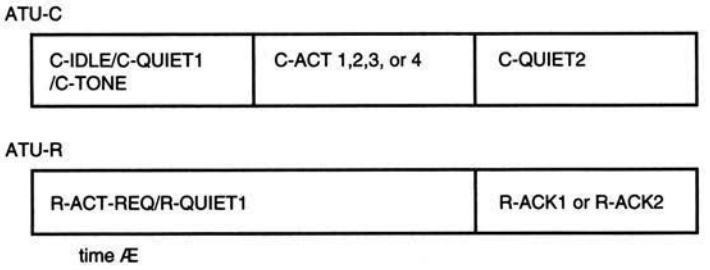
The whole initialization process takes about 24,000 symbols or, equivalently, six seconds of real-time. The initialization process has four phases:

1. *Activation and acknowledgment (384 symbols).* ATU-C and ATU-R detect each other and exchange information about timing and pilot tone options.
2. *Transceiver training (7,808 symbols).* Automatic gain adjustment, echo canceller, and channel equalizer training are performed.
3. *Channel analysis (18,186 symbols).* Rate options are indicated, and subchannel SNR analysis is performed.
4. *Exchange (about 200 symbols).* Exchange of information about loop attenuation, performance margin, and number of bits supported takes place. Bit loading selections are confirmed.

Activation and Acknowledgement Phase

Figure 13.74 illustrates phase one of initialization: activation and acknowledgment.

Figure 13.74 Activation and Acknowledgment Phase



R-ACT-REQ is a single sinusoid with a frequency of 34.5 kHz corresponding to upstream tone #8. The tone should be transmitted for 1024 symbols.

C-ACT1 is a single sinusoid with a frequency of 207 kHz corresponding to downstream tone #48. C-ACT2 is a single sinusoid with a frequency of 189.75 kHz corresponding to downstream tone #44. C-ACT3 is a single sinusoid with a frequency of 224.25 kHz corresponding to downstream tone #52. C-ACT4 is a single sinusoid with a frequency of 258.75 kHz corresponding to downstream tone #60. These C-ACTs should last for 128 symbols without a cyclic prefix.

R-ACK1 is a single sinusoid with a frequency of 43.125 kHz corresponding to upstream tone #10. R-ACK2 is a single sinusoid with a frequency of 34.5 kHz corresponding to upstream tone #12. R-ACK3 is a single sinusoid with a frequency of 60.375 kHz corresponding to upstream tone #14. R-ACK3 is still reserved for future initialization options. R-ACK should last 128 symbols.

Transceiver Training Phase

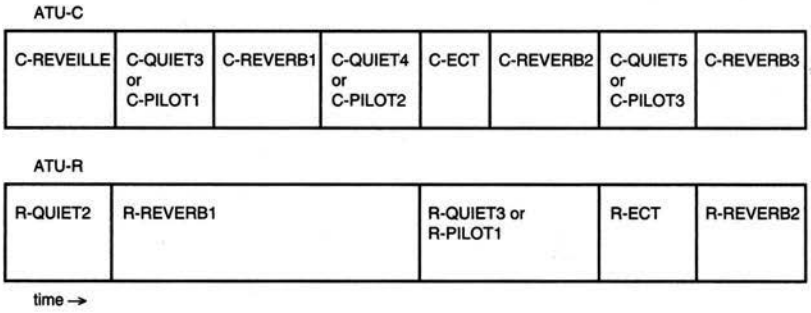
Figure 13.75 illustrates phase two of initialization: transceiver training.

C-REVEILLE is a sinusoid with a frequency of 241.5 kHz corresponding to downstream tone #56. C-REVEILLE lasts 128 symbols. The R-REVERB1 uses the same pseudo random sequence as that used for upstream synchronization symbol, but without cyclic prefix. The pseudo random sequence is repeated 64 times for a duration of 4096 symbols.

C-PILOT1 or C-PILOT2 is a sinusoid with a frequency of 276 kHz corresponding to downstream tone #64. C-PILOT1 or C-PILOT2 lasts at least 512 symbols. The C-REVERB1 uses the same pseudo random sequence as that used for downstream

synchronization symbol but without cyclic prefix. The C-REVERB1 allows the ATU-R to adjust its gain control.

Figure 13.75 Transceiver Training Phase



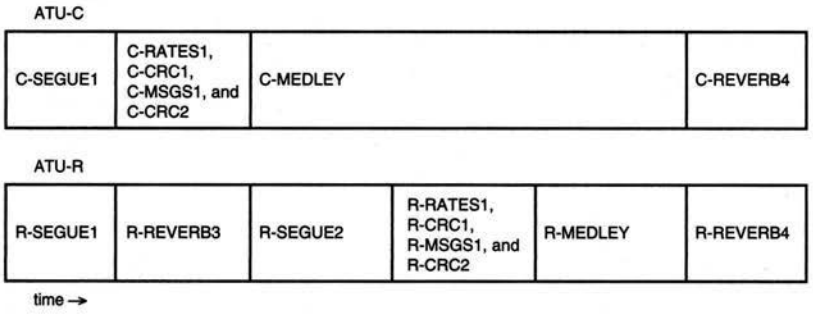
R-PILOT1 is a sinusoid with a frequency of 69 kHz corresponding to upstream tone #16. The R-PILOT1 should last 2048 symbols. C-ECT is a vendor-defined signal that is used by the echo canceller at ATU-C. The duration for C-ECT is 512 symbols. C-REVERB2 allows the ATU-R to perform synchronization and to train its equalizer. C-REVERB2 has the same signal format as C-REVERB1, but lasts for 1536 symbols.

C-PILOT3 is the same as C-PILOT1. C-REVERB3 is the same as C-REVERB2. C-REVERB3 allows the ATU-R to perform or maintain synchronization and to further train its equalizer. R-ECT is also a vendor-defined signal for training the echo canceller at ATU-R. R-ECT also has a duration of 512 symbols. R-REVERB2 is used by ATU-C to perform timing recovery and equalizer training. R-REVERB2 is the same as R-REVERB1, but lasts only 1024 symbols.

Channel Analysis Phase

Figure 13.76 illustrates phase three of initialization: channel analysis.

Figure 13.76 Channel Analysis Phase



C-SEGUE1 is generated from a tone-by-tone 180 degree phase reversal of C-REVERB1. R-SEGUE1 is also generated from a tone-by-tone 180 degree reversal of R-REVERB1. C-RATES is the first ATU-C signal for which a cyclic prefix is used. The purpose of C-RATES1 is to transmit four options for data rates and formats to the ATU-R. The duration of C-RATES1 is 992 symbols. C-CRC1 is a cyclic redundancy code for the detection of errors in the reception of C-RATES1. The duration of C-CRC1 is 16 symbols.

C-MSGS1 transmits a 48-bit message signal to ATU-R. This message includes vendor identification, ATU-C transmit power level used, Trellis code option, echo cancellation option, and so on. The duration of C-MSGS1 is 48 symbols. C-CRC2 is a 16-bit cyclic redundancy code for the detection of errors in the reception of C-MSGS1. C-CRC2 also has a duration of 16 symbols. R-REVERB3 is similar to R-REVERB1, but with the addition of cyclic prefix. The duration of R-REVERB3 can have a maximum of 4000 symbols. R-REVERB3 should last 20 symbols longer than the C-CRC2.

C-MEDLEY is used by ATU-R for estimation of downstream SNR. C-MEDLEY uses the same pseudo random sequence as that used by C-REVERB1 except no 1s are filled at every 512 positions. Therefore, each subchannel sees a different signal from one symbol period to the next symbol period. C-MEDLEY lasts for 16,384 symbols. R-SEGUE2 is the same signal as R-SEGUE1 for 10 symbol periods.

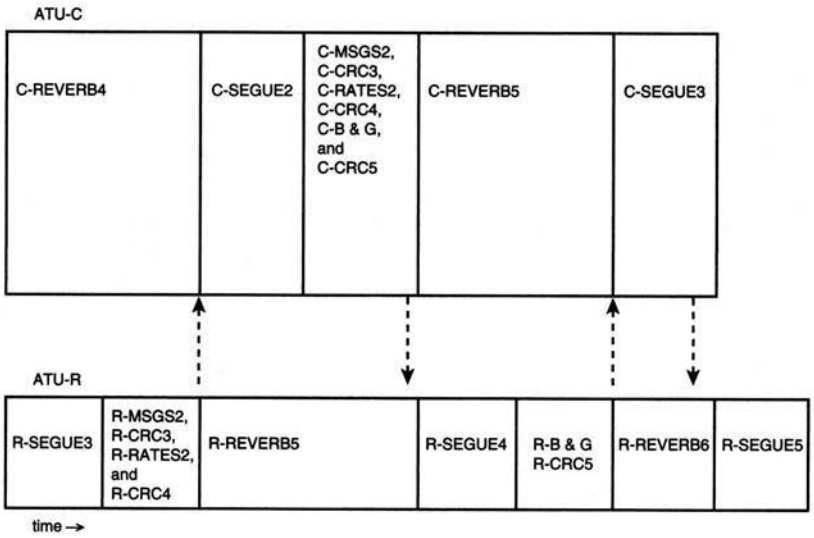
Exchange Phase

Figure 13.77 illustrates phase four of initialization: exchange.

R-SEGUE3 is the same as R-SEGUE1. The duration of R-SEGUE3 is 10 symbol periods. R-MSGS2 transmits a 32-bit message signal to the ATU-C during a four symbol period. Two bits are encoded onto each of the sub-carriers numbered 6 through 9 using the 4QAM constellation. This message includes the total number of bits per symbol supported, the estimated upstream loop attenuation, and the performance margin with the selected rate option. R-CRC3 is a cyclic redundancy code for the detection of errors in the reception of R-MSGS2 by the ATU-C. R-CRC3 lasts two symbol periods.

R-RATES2 is the reply to C-RATES1, based on the results of the downstream channel analysis. The ATU-R sends back only the option number of the highest data rate that can be supported based on the measured SNR of the downstream channel. A total of eight bits are used for R-RATES2 in one symbol period. R-CRC4 is a cyclic redundancy code for the detection of errors in the reception of R-RATES2 at the ATU-C. R-CRC4 also lasts two symbol periods.

Figure 13.77 Exchange Phase



C-MSGS2 transmits a 32-bit message to the ATU-R. This message includes the total number of bits per symbol supported, the estimated upstream loop attenuation, and the performance margin with selected rate option. The C-MSGS2 lasts four symbol periods. C-CRC3 is a cyclic redundancy code for the detection of errors in the reception of C-MSGS2 for the ATU-R. The C-CRC3 lasts two symbol periods. C-RATES2 is the reply to R-RATES1.

C-RATES2 combines the downstream rate information contained in R-RATES2 with the option number of the highest upstream data rate that can be supported based on the measured SNR of the upstream channel. It thus transmits the final decision on the rates that will be used in both directions. It lasts one symbol period. C-CRC4 is the cyclic redundancy code for the detection of errors in the reception of C-RATES2 at the ATU-R. C-B&G transmits the bits and gains that are to be used on the upstream carriers to the ATU-R. A total of 62 bytes of bits and gains information is transmitted during 62 symbol periods. C-CRC5 is a cyclic redundancy code for the detection of errors in the reception of C-B&G at the ATU-R.

C-REVERB5 is the same as C-REVERB4. The ATU-C should transmit C-REVERB5 until it has received, checked the reliability of, and established the downstream bits and gain information contained in R-B&G. The purpose of R-B&G is to transmit the bits and gains information to be used on the downstream sub-carriers to the ATU-C. A total of 510 bytes

of bits and gains information is transmitted during 510 symbol periods. R-CRC5 is a cyclic redundancy code for the detection of errors in the reception of R-B&G at the ATU-C.

R-REVERB6 is the same as R-REVERB3. The ATU-R transmits R-REVERB6 until it detects all ten symbols of C-SEGUE3. The maximum duration for R-REVERB6 is 4000 symbols. The ATU-C switches from sending C-REVERB5 to sending C-SEGUE3 after receiving, checking the reliability of, and establishing the downstream bits and gains information contained in R-B&G. C-SEGUE3 notifies the ATU-R that the ATU-C is about to enter the steady-state state C-SHOWTIME. The signal of C-SEGUE3 is the same as C-SEGUE1. The duration of C-SEGUE3 is 10 symbol periods. The ATU-R switches from sending R-REVERB6 to sending R-SEGUE5 after detecting 10 symbols of C-SEGUE3. The purpose of R-SEGUE5 is to notify the ATU-C that the ATU-R is about to enter the steady-state state of R-SHOWTIME. R-SEGUE5 lasts 10 symbol periods.

End Notes

1. D. L. Waring, "The Asymmetrical Digital Subscriber Line (ADSL): A New Transport Technology for Delivering Wideband Capacities to the Residence," *Proc. IEEE GLOBECOM '91*: 1979–1986.

2. J. L. Lechleider, "Asymmetrical Digital Subscriber Lines," *Bellcore Internal Technical Memorandum* (September 4, 1989).

3. E. E. Manchester and L. D. Hofer, "T1 Standards Project Proposal for an Asymmetrical Digital Subscriber Line (ADSL)," T1E1.4/91-079 (May 13, 1991).

4. K. Sistanizadeh, "Proposed Spectrum Management Guide Lines for ADSL Deployment Using Single-Tone QAM," *Bellcore T1E1 Contribution,* T1E1.4/91-121 (August 1991).

5. K. Sistanizadeh, "Spectral Compatibility of Qam ADSl with Basic Rate DSL," *Bellcore T1E1 Contribution,* T1E1.4/91-118 (August 1991).

6. K. Sistanizadeh, "Proposed Canonical Loops for ADSL and Their Loss Characteristics," *Bellcore T1E1 Contribution,* T1E1.4/91-116 (August 1991).

7. K. J. Kerpez, "The Error Rate of ADSL in the Presence of Impulse Noise," *Bellcore T1E1 Contribution,* T1E1.4/91-122 (August 1991).

8. J. Cioffi and K. Maxwell, "A Comparison of Multi-Carrier and Single-Carrier Modulation for ADSL," *Amati T1E1 Contribution,* T1E1.4/91-123 (August 1991).

9. J. M. Cioffi, "A Multicarrier Primer," *Stanford University/Amati T1E1 Contribution,* T1E1.4/91-157 (November 1991).

10. J. Cioffi, P. Chow, and R. K. Maxwell, "Range Limits for Multicarrier DSL Systems," *Stanford University/Amati T1E1 Contribution,* T1E1.4/158 (November 1991).

11. P. S. Chow, J. Cioffi, and R. K. Maxwell, "Performance of Multicarrier with DSL Impulse Noise," *Stanford University/Amati T1E1 Contribution,* T1E1.4/158 (November 1991).

12. K. J. Kerpez, "A Model for the Probability Distribution of the Length of Burst Errors Caused by Impulse Noise," *Bellcore T1E1 Contribution,* T1E1.4/91-165 (November 1991).

13. K. J. Kerpez, "Forward Error Correction Codes for ADSL," *Bellcore T1E1 Contribution,* T1E1.4/91-166 (November 1991).

14. E. Arnon and E. J. Weldon, "Error Correction Codes in ADSL," *Northern Telecom T1E1 Contribution,* T1E1.4/91-179 (November 1991).

15. W. Y. Chen, "An ADSL Control Channel Architecture," *Bellcore T1E1 Contribution,* T1E1.4/91-164 (November 1991).

16. P. T. Tong, J. T. Aslanis, and J. M. Cioffi, "A Trellis-Code Proposal for ADSL Line Signals," *Amati T1E1 Contribution,* T1E1.4/91-176 (August 1992).

17. W. Y. Chen, "Carrierless Generation and Reception of Passband Signals for ADSL," *Bellcore T1E1 Contribution,* T1E1.4/92-024 (February 1992).

18. K. Sistanizadeh, "Discrete Multi-Tone Signaling (DMT) and Single-Tone Quadrature Amplitude Modulation (QAM) Performance for ADSL," *Bellcore T1E1 Contribution,* T1E1.4/92-026 (February 1992).

19. J. M. Cioffi, P. S. Chow, and R. K. Maxwell, "Multicarrier Range and Performance for ADSL-I (1.6 Mbps) and ADSL-II (4 Mbps)," *Amati T1E1 Contribution,* T1E1.4/92-007 (February 1992).

20. L. Ebringer, "Spectral Compatibility of QAM ADSL with HDSL and Basic Rate DSL," *Cambridge University T1E1 Contribution,* T1E1.4/92-027 (February 1992).

21. B. Daneshrad and H. Samueli, "Performance Analysis of a QAM Adaptive Receiver for 1.6 Mbps Digital Subscriber Line Transmission," *UCLA T1E1 Contribution,* T1E1.4/92-037 (February 1992).

22. M. Sorbara, "System Definition for ADSL," *AT&T T1E1 Contribution,* T1E1.4/92-066 (May 1992).

23. J. M. Cioffi and R. K. Maxwell, "A Programmable Protocol for ADSL Transceivers," *Amati T1E1 Contribution,* T1E1.4/92-076 (May 1992).

24. R.C. McConnell, "ADSL, ADSLII, and ADSLIII: Interfaces & Example Functional Architectures," *Bellcore T1E1 Contribution,* T1E1.4/92-083 (May 1992).

25. K. Swaminathan and M. R. Brown, "ADSL Operations Issues and Discussion Questions," *Bellcore T1E1 Contribution,* T1E1.4/92-084 (May 1992).

26. R.C. McConnell, "ADSL Overhead Function Bandwidth Allocations and Example Binary Equivalent Line Rates," *Bellcore T1E1 Contribution,* T1E1.4/92-085 (May 1992).

27. M. Sorbara, "ADSL Frame Format," *AT&T T1E1 Contribution,* T1E1.4/92-092 (May 1992).

28. K. R. Maxwell, "ADSL Service Options," *T1E1 Contribution,* T1E1.4/92-104 (May 1992).

29. K. Sistanizadeh, "A 16-QAM 16 kb/s Reverse ADSL Control Channel Spectrum Allocation and Performance," *Bellcore T1E1 Contribution,* T1E1.4/92-077 (May 1992).

30. K. Sistanizadeh, "Spectral Compatibility of a 16/64 QAM ADSL with Basic Rate DSL-Induced NEXT and AWGN," *Bellcore T1E1 Contribution,* T1E1.4/92-079 (May 1992).

31. M. Hoque, "Electromagnetic Interference Performance of QAM ADSL," *Bellcore T1E1 Contribution,* T1E1.4/92-080 (May 1992).

32. L. Baker, "Intra System Impulse Noise in ADSL/POTS System," *R-TEC T1E1 Contribution,* T1E1.4/92-070 (May 1992).

33. K. Sistanizadeh, "A Test Plan Proposal for Spectral Compatibility Evaluation of ADSL Transceiver Prototypes," *Bellcore T1E1 Contribution,* T1E1.4/92-081 (May 1992).

34. W. Y. Chen, "A Calculation of the Required A/D Precision for ADSL," *Bellcore T1E1 Contribution,* T1E1.4/92-082 (May 1992).

35. A. Hollbach, "Application and Service Drivers for an ADSL Standards Project in TIEI," *NTI T1E1 Contribution,* T1E1.4/92-107 (May 1992).

36. C. Valenti, "Standards Project Proposal: Interfaces Relating to Carrier to Customer Connection of Asymmetric Digital Subscriber Line (ADSL) Equipment," *Bellcore T1E1 Contribution,* T1E1.4/92-086 (May 1992).

37. K. J. Kerpez, "The Range of Baseband ADSLs as a Function of Nit Rate," *Bellcore T1E1 Contribution,* T1E1.4/91- (August 1991).

38. W. Y. Chen, "Asymmetrical Echo Cancellation for a Baseband ADSL," *Bellcore T1E1 Contribution,* T1E1.4/92-156 (August 1992).

39. J. Lechleider, "Impulse Noise Cancellation for ADSL," *Bellcore T1E1 Contribution,* T1E1.4/92-154 (August 1992).

40. W. Y. Chen, G. H. Im, and J. J. Werner, "Design of Digital Carrierless AM/PM Transceivers," *AT&T and Bellcore T1E1 Contribution,* T1E1.4/92-149 (August 1992).

41. H. Samueli, R. Joshi, and B. Daneshrad, "VLSI Complexity Comparison of a QAM vs. CAP ADSL Transceiver," *UCLA T1E1 Contribution,* T1E1.4/92-144 (August 1992).

42. C. Valenti, "Standards Project Proposal: Interfaces Relating to Carrier to Customer Connection of ADSL Equipment," *Bellcore T1E1 Contribution,* T1E1.4/92-086R2 (August 1992).

43. J. M. Cioffi, J. T. Aslanis, and J. A. C. Bingham, "A DMT Proposal for ADSL Transceiver Interfaces," *Amati T1E1 Contribution,* T1E1.4/92-174 (August 1992).

44. K. T. Foster and J. W. Cook, "A Symbolic Pulse for Impulsive Noise Testing," *BTL T1E1 Contribution,* T1E1.4/92-143 (August 1992).

45. B. A. Blake, "Generating Crosstalk Signals for Testing ADSL Systems," *Bellcore T1E1 Contribution,* T1E1.4/92-148 (August 1992).

46. D. L. Waring and K. Sistanizadeh, "Suggested Plan to Access and Select ADSL Transmission Schemes for Further Study," *Bellcore T1E1 Contribution,* T1E1.4/92-147 (August 1992).

47. E. Arnon and M. Elder, "ADSL, HDSL, and BRA Powers and Spectrum Compatibility," *NTI T1E1 Contribution,* T1E1.4/92-169 (August 1992).

48. E. Arnon and M. Elder, "BRA-ISDN and ADSL Signals on the Same Loop," *NTI T1E1 Contribution,* T1E1.4/92-170 (August 1992).

49. J. T. Aslanis, N. Al-Dhahir, and J. A. C. Bingham, "Performance of the Proposed DMT Line Code for ADSL with Embedded ISDN," *Amati T1E1 Contribution,* T1E1.4/92-178 (August 1992).

50. K. J. Kerpez, "Coding for ADSL," *Bellcore T1E1 Contribution,* T1E1.4/92-153 (August 1992).

51. P. T. Tong, J. T. Aslani, and J. M. Cioffi, "A Trellis-Code Proposal for ADSL Line Signals," *Amati T1E1 Contribution,* T1E1.4/92-176 (August 1992).

52. H. L. Lou, P. T. Tong, and J. M. Cioffi, "A Programmable VLSI Codec for Trellis-Coding," *Amati T1E1 Contribution,* T1E1.4/92-177 (August 1992).

53. J. T. Aslanis, P. T. Tong, and T. N. Zogakis, "An ADSL Proposal for Selectable Forward Error Correction with Convolution Interleaving," *Amati T1E1 Contribution,* T1E1.4/92-176 (August 1992).

54. D. L. Waring and K. Sistanizadeh, "A Revised Suggested Plan to Access ADSL Transmission Schemes for Further Study," *Bellcore T1E1 Contribution,* T1E1.4/92-147R2 (December 1992).

55. N. Epstein, R. Olshansky, and R. Gross, "ADSL Test Plan," *GTE T1E1 Contribution,* T1E1.4/92-217 (December 1992).

56. P. T. Tong, H. L. Lou, and P. H. Ang, "An Update on the Programmable Trellis Codes," *Amati and LSI Logic T1E1 Contribution,* T1E1.4/92-200 (December 1992).

57. T. N. Zogakis and J. T. Aslanis, "The Implementation of Selectable FEC for ADSL," *Amati T1E1 Contribution,* T1E1.4/92-202 (December 1992).

58. K. Manghani, "Availability of FEC and Related Chips for ADSL," *LSI Logic T1E1 Contribution,* T1E1.4/92-218 (December 1992).

59. K. Hohhof, "Forward Error Correction Proposal for ADSL," *Westell T1E1 Contribution,* T1E1.4/92-208 (December 1992).

60. G. Young, "Impulsive Noise in ADSL Systems," *B T Labs T1E1 Contribution,* T1E1.4/92-214 (December 1992).

61. J. S. Chow and R. R. Hunt, "Impulse Noise Measurements for DMT System," *Amati T1E1 Contribution,* T1E1.4/92-211 (December 1992).

62. K. J. Kerpez, "Minimum Mean Squared Error Impulse Noise Modeling," *Bellcore T1E1 Contribution,* T1E1.4/92-226 (December 1992).

63. W. Y. Chen, "An Analytical Test Bed for QAM/CAP ADSL Transceivers," *Bellcore T1E1 Contribution,* T1E1.4/92-224 (December 1992).

64. J. A. C. Bingham, "Design of Multitone Transceivers," *Amati T1E1 Contribution,* T1E1.4/92-198 (December 1992).

65. J. Cioffi, J. Aslanis, and M. Ho, "Performance of Enhanced (6 Mbps) ADSL," *Amati and Stanford University T1E1 Contribution,* T1E1.4/92-205 (December 1992).

66. T. Starr, "ADSL Bit Rates," *Ameritech T1E1 Contribution,* T1E1.4/92-231 (December 1992).

67. P. J. Kyees, "Suggestion to Remove Bit-Rate-Limit of Payload on Initial Work Outlined in ADSL Standards Project Proposal," *BellSouth Telecommunications T1E1 Contribution,* T1E1.4/92-246 (December 1992).

68. M. Ho, J. Cioffi, and J. A. C. Bingham, "An Echo Cancellation Method for DMT with DSLs," *Amati and Stanford University T1E1 Contribution,* T1E1.4/92-201 (December 1992).

69. H. Y. Chung, G. H. Im, M. Sorbara, and J. J. Werner, "Spectrum Compatibility Issues for DSLs," *AT&T T1E1 Contribution,* T1E1.4/92-233 (December 1992).

70. K. Sistanexadeh, "Spectral Compatibility of 2B1Q HDSL with 16-QAM ADSL Induced NEXT and AWGN," *Bellcore T1E1 Contribution,* T1E1.4/92-222 (December 1992).

71. J. Cioffi, "ADSL Maintenance with DMT," *Amati T1E1 Contribution,* T1E1.4/92-203 (December 1992).

72. K. Hohhof and R. Koval, "Frame Structure Proposal for ADSL," *Westell T1E1 Contribution,* T1E1.4/92-209 (December 1992).

73. J. Aslanis, J. Cioffi, and R. M. Adolf, "T1.601 Compatible Full ADSL Service with POTS," *Amati T1E1 Contribution,* T1E1.4/92-199 (December 1992).

74. K. J. Kerpez, "An Update on the Recommended Reed-Solomon Code for ADSL," *Bellcore T1E1 Contribution,* T1E1.4/93-035 (March 1993).

75. T. N. Zogakis, P. T. Tong, and J. Cioffi, "Performance Comparison of FEC/Interleave Choices with DMT for ADSL," *Amati T1E1 Contribution,* T1E1.4/93-091 (March 1993).

76. J. T. Aslanis, J. S. Chow, and P. T. Tong, "A Selective Error Correction Proposal for ADSL," *Amati T1E1 Contribution,* T1E1.4/93-023 (March 1993).

77. J. Cioffi and J. A. C. Bingham, "Echo Cancellation for ADSL," *Amati T1E1 Contribution,* T1E1.4/93-091 (March 1993).

78. J. Cioffi, "Revisiting Recommended DMT Line Code," *Amati T1E1 Contribution,* T1E1.4/93-021 (March 1993).

79. J. S. Chow and J. Cioffi, "DMT Initialization: Parameters Needed for Specification in a Standard," *Amati T1E1 Contribution,* T1E1.4/93-022 (March 1993).

80. J. Cioffi and P. T. Tong, "VLSI DMT Implementation for ADSL," *Amati T1E1 Contribution,* T1E1.4/93-025 (March 1993).

81. C. T. Throop, "Recommendation for the Inclusion of 6 Mbps Payload in the Initial Standard," *ECI Telecom, Inc., T1E1 Contribution,* T1E1.4/93-052 (March 1993).

82. L. M. Smith and M. Sorbara, "Alternative 6 Mbps ADSL Architectures: Advantages and Disadvantages," *AT&T T1E1 Contribution,* T1E1.4/93-055 (March 1993).

83. W. Y. Chen, "The Effect of ADSL Spectrum on T1 Performance," *Bellcore T1E1 Contribution,* T1E1.4/93-033 (March 1993).

84. E. Arnon and M. Elder, "Performance Measurement of ADSL System," *NTI T1E1 Contribution,* T1E1.4/93-064 (March 1993).

85. B. A. Blake, "Description of the Transmission Tests Performed on ADSL Transceiver Prototypes at Bellcore," *Bellcore T1E1 Contribution,* T1E1.4/93-029 (March 1993).

86. B. A. Blake, "Results of Transmission Tests on an ADSL Transceiver Prototype from Reliance COMM/TEC and Bellcore," *Bellcore T1E1 Contribution,* T1E1.4/93-030 (March 1993).

87. B. A. Blake, "Results of Transmission Tests on an ADSL Transceiver Prototype from Amati Communications Corp.," *Bellcore T1E1 Contribution,* T1E1.4/93-031 (March 1993).

88. B. A. Blake, "Results of Transmission Tests on an ADSL Transceiver Prototype from AT&T Paradyne," *Bellcore T1E1 Contribution,* T1E1.4/93-032 (March 1993).

89. J. S. Chow, J. C. Tu, and J. M. Cioffi, "A Discrete Multitone Transceiver System for HDSL Applications," *IEEE JSAC,* vol. 9, no. 6 (August 1991): 895–908.

90. S. B. Weistain and P. M. Ebert, "Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform," *IEEE Trans. Commun. Tech.,* vol. COM-19 (October 1971): 628–634.

91. J. S. Chow and J. M. Cioffi, "A Cost-Effective Maximum Likelihood Receiver for Multicarrier Systems," *Proc. IEEE ICC '92* (June 1992): 948–952.

92. J. S. Chow, J. M. Cioffi, and J. A. C. Bingham, "Equalizer Training Algorithm for Multicarrier Modulation Systems," *Proc. IEEE ICC '93* (June 1993): 761–765.

93. J. C. Tu and J. M. Cioffi, "A Loading Algorithm for the Concatenation of Coset Codes with Multichannel Modulation Methods," *Proc. IEEE Globecom* (1990): 703.6.1–703.6.5.

94. J. T. Aslanis, N. Al-Dhahir, and J. A. C. Bingham, "Performance of the Proposed DMT Line Code for ADSL with Embedded ISDN," *Amati T1E1 Contribution,* T1E1.4/92-178 (August 1992).

95. P. T. Tong, J. T. Aslani, and J. M. Cioffi, "A Trellis-Code Proposal for ADSL Line Signals," *Amati T1E1 Contribution,* T1E1.4/92-176 (August 1992).

96. J. T. Aslanis, P. T. Tong, and T. N. Zogakis, "An ADSL Proposal for Selectable Forward Error Correction with Convolution Interleaving," *Amati T1E1 Contribution,* T1E1.4/92-180 (August 1992).

97. P. S. Chow, J. Cioffi, and R. K. Maxwell, "Performance of Multicarrier with DSL Impulse Noise," *Stanford University/Amati T1E1 Contribution,* T1E1.4/158 (November 1991).

98. K. J. Kerpez, "A Model for the Probability Distribution of the Length of Burst Errors Caused by Impulse Noise," *Bellcore T1E1 Contribution,* T1E1.4/91-165 (November 1991).

99. K. J. Kerpez, "Forward Error Correction Codes for ADSL," *Bellcore T1E1 Contribution,* T1E1.4/91-166 (November 1991).

100. S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications* (Prentice-Hall, 1983).

101. J. M. Cioffi and J. A. C. Bingham, "A Data-Driven Multitone Echo Canceller," *IEEE Trans. On Commun.,* vol. 42, no. 10 (October 1994): 2853–2869.

102. M. Ho, J. M. Cioffi, and J. A. C. Bingham, "Discrete Multitone Echo Cancellation," *IEEE Trans. On Commun.,* vol. 44, no. 7 (July 1996): 817–825.

103. J. Cioffi and P. T. Tong, "VLSI DMT Implementation for ADSL," *Amati T1E1 Contribution,* T1E1.4/93-025 (March 1993).

104. W. Y. Chen, G. H. Im, and J. J. Werner, "Design of Digital Carrierless AM/PM Transceivers," *AT&T and Bellcore T1E1 Contribution,* T1E1.4/92-149 (August 1992).

105. M. Tomlinson, "New Automatic Equalizer Employing Modulo Arithmatic," *Electron. Lett.,* vol. 7 (March 1971): 138–139.

106. B. A. Blake, "Results of Transmission Tests on an ADSL Transceiver Prototype from AT&T Paradyne," *Bellcore T1E1 Contribution,* T1E1.4/93-032 (March 1993).

107. D. Amrany, S. Gadot, and M. Dimyan, "A Programmable DSP Engine for High-Rate Modems," *Proc. IEEE ISSCC '92* (June 1992): 222–223.

108. W. Y. Chen and D. L. Waring, "ADSL Noise Environment and Potential Performance," *Bellcore Internal Technical Memorandum* (September 23, 1993).

109. J. C. Tu and J. M. Cioffi, "A Loading Algorithm for the Concatenation of Coset Codes with Multichannel Modulation Methods," *Proc. IEEE Globecom* (1990): 703.6.1–703.6.5.

110. W. Y. Chen, "Architecture and Performance Simulation of Single-Pair HDSL," *Proc. IEEE ICC '96* (June 1996).

111. D. L. Waring, "The Asymmetrical Digital Subscriber Line (ADSL): A New Transport Technology for Delivering Wideband Capacities to the Residence," *Proc. IEEE GLOBECOM '91*: 1979–1986.

112. British Telecom Contribution.

113. IEEE P1394 standards.

CHAPTER 14

SHDSL (Single-Pair High-Bit-Rate Digital Subscriber Lines)

The technical report of HDSL served the purpose of an industry standard interface for both equipment manufacturers and telephone companies. With additional effort on details of equipment interoperability, HDSL enjoyed a healthy market share as a dual-duplex repeaterless T1 technology. HDSL semiconductor chips with reliable performance are available from many vendors. Competition for the market share has lead to a higher level of semiconductor circuit integration and some technical innovations. It is expected that with more signal processing power plus the help of advanced coding techniques, the transmission throughput of HDSL can be realized on a single pair of CSA range loops.

At the June 1995 T1E1.4 meeting, Metalink made a proposal for a *single-pair HDSL* (SHDSL) study project.[1] It was argued that the single-pair HDSL had been offered as a product and a technical report should be generated to help the convergence of an industry standard. It was claimed that the 6 dB performance margin could be achieved on five CSA test loops and the 6 dB margin reaches were 10,800 feet for the 24-AWG loop and 7,800 feet for the 26-AWG loop.[2] The performance is achieved without any coding gain, but a proprietary Near Maximum Likelihood (NML) algorithm was used. The similar performance was also considered achievable by a CAP system with some coding gain.[3]

After that initial discussion, the single-pair HDSL (SHDSL) iteration has been an active discussion topic during subsequent T1E1.4 meetings. SHDSL is also referred as *HDSL2.* Most recently, the term *Next Generation HDSL* has been used for activities related to establishing an SHDSL standard. This chapter reviews major discussion topics related to T1E1.4 working group SHDSL activities. The current status of a possible SHDSL standard is also discussed. This chapter covers a myriad of SHDSL related topics including the following:

- *Performance improvements.* Measured and estimated performance margins as well as transmission throughputs on a single-pair CSA loop with or without coding are reviewed. Channel capacities of CSA loops are calculated in the context of SHDSL. The effect of the NEXT model and the issue of spectrum compatibility are addressed.
- *Coding gain.* Trellis codes with up to 1024 states are examined in conjunction with a Viterbi decoder or a sequential decoder based on the Fano algorithm. The applicability of Turbo codes is also discussed.
- *Minimizing NEXT with spectrum shaping.* Proposals of symmetrical and asymmetrical signal spectra and associated performance estimates are reviewed.
- *HDSL2 standard status.* The convergence of an SHDSL standard is discussed.

14.1 The Desired Performance Improvements

Metalink presented measurement results on CSA loops and 24-AWG and 26-AWG loops. For CSA loops #1, #4, #6, #7, and #8, the length of the longest section is made variable such that by reducing the variable section loop length, a performance margin (above the 10^{-7} bit error rate requirement) of 6 dB can be achieved. The percentage reach of the CSA loop is then calculated by dividing the total loop length (defined as the sum of lengths of fixed sections plus the length of the variable section) by the CSA loop length. The lengths of bridged taps are not included in the percentage calculation. Table 14.1 shows the CSA loop measurement results for both forward (from CO to subscriber) and reverse (Rev.) directions.

Table 14.1 Reach of a Single-Pair HDSL System

Loop	Variable Length (feet)	Total Loop Length (feet)	CSA Loop Length (feet)	Percentage of the CSA Loop
#1	5000	6900	7700	89
#1 Rev.	5000	6900	7700	89
#4	4550	5860	7600	77
#4 Rev.	4550	5810	7600	76
#6	7800	7800	9000	86
#7	10000	10000	10700	93
#7 Rev.	9200	9150	10700	86
#8	10600	10600	12000	88

Figures 14.1 and 14.2 show the performance results of a single-pair HDSL system on 24-AWG and 26-AWG loops, respectively.

Figure 14.1 Metalink Margin Over 24-AWG Loop

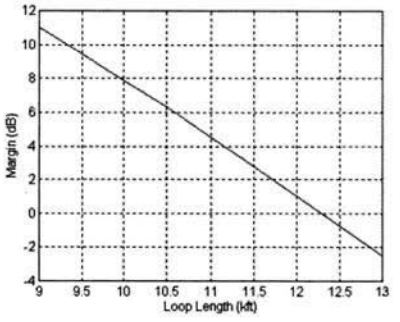
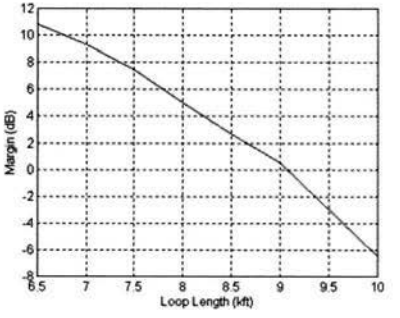


Figure 14.2 Metalink Margin Over 26-AWG Loop



14.1.1 EC/FDM-Based CAP SHDSL Systems

Performances of an Echo Cancellation (EC)–based system and a Frequency Division Multiplexed (FDM)–based system were estimated by AT&T's contribution to the June 1995 T1E1.4 meeting. The EC-based system has a CAP 64 line code with a 3 dB bandwidth of 266 kHz and a 15-percent excessive bandwidth of 368 kHz. Because CAP is a passband system, the lower-corner frequency can be located at either 0 Hz or 35 kHz. For the latter case, a POTS channel can be made available simultaneously. The FDM system also has a CAP 64 line code. The upstream channel is located from 35 kHz to 403 kHz and the downstream channel is located from 580 kHz to 948 kHz. In the performance estimation, a 4 dB Trellis coding gain is assumed for both EC- and FDM-based systems.

The performance margin is about 1.5 dB for the EC system with 0 Hz lower-corner frequency on the 9 kft 26-AWG loop. The 6 dB performance margin reach is about 8 kft for the 26-AWG loop and 11.5 kft for the 24-AWG loop. When the lower-corner frequency is raised to 35 kHz, the performance margin is reduced to –1.1 dB on the 9 kft 26-AWG loop. The 6 dB performance margin reach is reduced to 7.3 kft for the 26-AWG loop and 10.3 kft for the 24-AWG loop. On the other hand, performance margins of the FDM-based system are estimated to be 11.3 dB and 13 dB on the 9 kft, 26-AWG loop and the 12 kft, 24-AWG loop, respectively.

14.1.2 DMT Line Code for SHDSL Systems

It is also possible to use the DMT[4] line code for the single-pair HDSL. The challenge for the DMT system is the delay constraint. To reduce the amount of *Inverse Fast Fourier Transform* (IFFT) and *Fast Fourier Transform* (FFT) delay, a small FFT size might be preferable. The percentage of cyclic prefix overhead, however, might increase unless a better time domain equalizer is adapted. The possible FFT sizes are 64 or 128 with a time domain equalizer length of up to 32 taps. Depending on the FFT size and bandwidth, a DMT-based single-pair HDSL could be realized with the assistance of coding gain.

14.1.3 Comparison of PAM and CAP/QAM Line Codes

ADTRAN's T1E1 contribution compares PAM and CAP/QAM line code performance in a single-pair HDSL system with a transmission throughput of 1.6 Mbps.[5] For this particular comparison study, the PAM family consists of 4, 8, and 16 level line codes. The best modulation type from the PAM family is determined to be coded 16-PAM, achieving reaches of up to 10,600 feet and 7,700 feet on 24-AWG and 26-AWG, respectively.

Coded 64-CAP is determined as the best modulation type from the CAP/QAM family, achieving reaches of up to 10,300 feet and 7,400 feet on 24-AWG and 26-AWG, respectively. Test results show that, when considering worst-case disturber types, coded 16-PAM has a loop reach up to 800 feet greater than coded 64-CAP. When considering only self-NEXT, coded 16-PAM and coded 64-CAP perform comparably. Tables 14.2 to 14.4 summarize performance simulation results for coded and uncoded PAM/CAP SHDSL systems with a 6 dB margin as well as performance margins at CSA loops.

AT&T's T1E1 contribution also compares CAP/QAM and PAM line code performance in SHDSL systems.[6] When deploying the transceiver without POTS, the results show that with the incorporation of Trellis coding and Tomlinson precoding in all transceivers, CAP/QAM and PAM can achieve essentially the same reach, assuming 4 dB of Trellis coding gain and that the CAP transmit spectrum starts at 0 Hz. An analysis of the following simulation results confirms that CAP/QAM and PAM achieve essentially the same loop reach.

Table 14.2 Summary of Uncoded PAM/CAP Results with a 6 dB Margin

	All Disturber		Self-NEXT	
	24 AWG	26 AWG	24 AWG	26 AWG
PAM Type (Uncoded)	8-PAM	8-PAM	4-PAM	4-PAM
PAM Loop Reach	9000	6500	9000	6700
CAP Type (Uncoded)	64-CAP	64-CAP	32-CAP	16-CAP
CAP Loop Reach	8600	6200	9000	6500

Table 14.3 Summary of Coded PAM/CAP Results with a 6 dB Margin

	All Disturber		Self-NEXT	
	24 AWG	26 AWG	24 AWG	26 AWG
PAM Type (Coded)	16-PAM	16-PAM	16-PAM	8-PAM
PAM Loop Reach	10600	7700	10600	7700
CAP Type (Coded)	128-CAP	128-CAP	64-CAP	64-CAP
CAP Loop Reach	10300	7400	10500	7700

Table 14.4 The Margin at CSA Loops

	All Disturber		Self-NEXT	
	12 kft, 24 AWG	9 kft, 26 AWG	12 kft, 24 AWG	9 kft, 26 AWG
Coded 16-PAM	2.6 dB	1.4 dB	2.6 dB	1.4 dB
Coded 64-CAP	1.1 dB	0.2 dB	2.2 dB	1.3 dB

On 24-gauge wire, coded 16-PAM shows a reach of 10.8 kft with 6 dB margin and coded 64-CAP has a reach of 10.7 kft with a 6.1 dB margin.

On 26-gauge wire, coded 8-PAM shows a reach of 7.8 kft with 6.2 dB margin and coded 64-CAP shows a reach of 7.8 kft with a 6.1 dB margin.

On CSA loop #8 (12 kft 24-AWG), coded 16-PAM shows 3.1 dB of margin and coded 64-CAP shows 2.9 dB of margin.

On CSA loop #6 (9 kft 26-AWG), both coded 64-CAP and 16-PAM show 1.9 dB of margin.

Tables 14.5 to 14.8 summarize the performance simulation results for reaches of 24-AWG

and 26-AWG loops and performance margins for CSA loop #6 and #8, all with self-NEXT. The slight CAP performance difference from these two studies might be due to the difference in CAP lower-corner frequencies.

Table 14.5 The Reach with 6 dB Margin of 24-AWG and Self-NEXT

Line Code (w/TCM)	Distance (kft)	Margin (dB)
16-PAM	10.8	6.0
64-CAP	10.7	6.1
128-CAP	10.6	6.0
8-PAM	10.6	6.0
32-CAP	10.4	6.2
256-CAP	10.3	6.0

Table 14.6 The Reach with 6 dB Margin of 26-AWG and Self-NEXT

Line Code (w/TCM)	Distance (kft)	Margin (dB)
8-PAM	7.8	6.2
64-CAP	7.8	6.1
16-PAM	7.8	6.0
32-CAP	7.7	6.1
128-CAP	7.6	6.1
256-CAP	7.3	6.2

Table 14.7 The Margin on CSA #8 (12 kft, 24-AWG)

Line Code (w/TCM)	Margin (dB)
16-PAM	3.1
64-CAP	2.9
128-CAP	2.6
8-PAM	2.4
32-CAP	2.1
256-CAP	2.1

Table 14.8 The Margin on CSA #6 (9 kft, 26-AWG)

Line Code (w/ TCM)	Margin (dB)
64-PAM	1.9
16-CAP	1.9
8-PAM	1.9
32-CAP	1.5
128-CAP	1.4
256-CAP	0.7

Performances of different PAM line codes with or without coding are compared by Level One's T1E1 contribution.[7] It is concluded that in the presence of self-NEXT, 2B1Q (8-PAM) and 3B1O (16-PAM) perform about the same. In the presence of self-NEXT plus crosstalk from ADSL systems, however, coded 3B1O (16-PAM) achieves a higher performance than coded 2B1Q (8-PAM). In some cases, 3B1O outperforms 2B1Q by as much as 3 dB. Table 14.9 summarizes the performance margins for the 9 kft, 26-AWG loop.

Table 14.9 Performance Margins for the 9 kft, 26-AWG Loop

	24 Self-NEXT (dB)	49 Self-NEXT (dB)	24 Self-NEXT+25 (dB)	49 ADSL (dB)
2B1Q	0.2	–1.5	–2.6	–3.6
3B1O	0	–1.7	–1.8	–2.2
Coded 2B1Q	4	2.5	1.4	0.4
Coded 3B1O	4.2	2.3	2.2	1.8

14.1.4 Channel Capacity for SHDSL Systems

The channel capacity in the context of the SHDSL environment can be used as a reference for the searching of the desired transmission performance. The calculation of the channel capacity requires the knowledge of the channel transfer function, noise model, and specific representation of hardware impairments.

The self-NEXT channel capacity can be calculated using the following expression:[8]

$$C = \int_0^F \log_2\left(1 + \frac{|H(f)|^2}{|X(f)|^2}\right) df$$

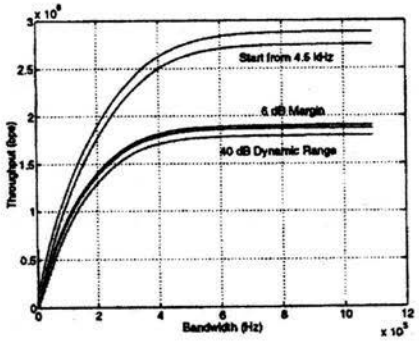
The values are as follows:

$H(f)$ is the loop transfer function.

$X(f)$ is the NEXT transfer function.

Figure 14.3 shows the channel capacity of the 9 kft, 26-AWG loop under the 49-disturber 1-percent worst-case scenario to be around 2.85 Mbps.

Figure 14.3 Channel Capacities for the 9 kft, 26-AWG Loop



The capacity is reduced to around 2.75 Mbps if the frequency below 4.5 kHz is not used for the general case of using a line coupling transformer. The channel capacity with a 6 dB performance margin, emulated with a 6 dB of noise level increase, is about 1.9 Mbps. The 6 dB performance margin channel capacity is reduced a little if the frequency below 4.5 kHz is not used. The 6 dB performance margin channel capacity is further reduced to about 1.75 dB if the system dynamic range is limited to 40 dB.

The transmission throughput of a DFE-based transceiver can also be displayed as a function of the signaling bandwidth (half of the baud rate) using the following approximate expression:

$$R = \frac{10\log_{10}\left(e^{T\int_0^{\frac{1}{T}} \ln\left(1+\sum_{n=-\infty}^{\infty}\frac{\left|H\left(f+\frac{n}{T}\right)\right|^2}{\left|X\left(f+\frac{n}{T}\right)\right|^2}\right)df}\right) - a}{bT}$$

The following are the values for an error rate of 10^{-7}:

$a = 9.5$

$b = 6$

The use of fractional number of bits per symbol is assumed possible.

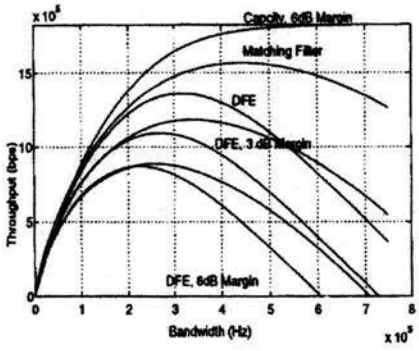
Similarly, the transmission throughput of a matching filter-based transceiver can be displayed as a function of the signaling bandwidth using the following approximate expression:

$$R = \frac{10\log_{10}\left(\frac{\int_0^{\frac{1}{2T}}|H(f)|^2 df}{\int_0^{\frac{1}{2T}}|X(f)|^2 df}+1\right) - a}{bT}$$

An ideal matching filter-based transceiver has a matching filter bank in front of the receiver to match any data symbol after the channel distortion. The performance of a matching filter-based transceiver is better because it does not cause much noise enhancement.

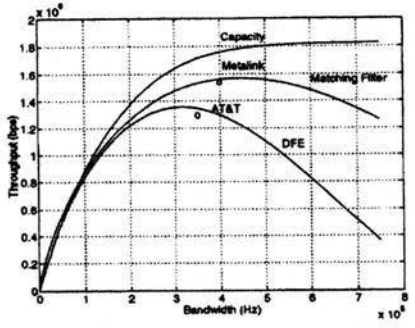
The achievable transmission throughput results are about 1.55 Mbps and 1.35 Mbps for a matching filter-based transceiver with a signaling bandwidth of 430 kHz and a DFE-based transceiver with a signaling bandwidth of 310 kHz, respectively, as shown in Figure 14.4. In other words, a coding gain of 6 dB is required for the matching filter-based transceiver with a 6 dB of performance margin and a larger than 6 dB of coding gain is required for a DFE-based transceiver with a 6 dB of performance margin. With only 3 dB of coding gain the achievable transmission throughput results are 1.2 Mbps and 1.1 Mbps, respectively. Transmission throughputs are below 900 kbps without any coding gain.

Figure 14.4 Channel Capacities for the 12 kft, 24-AWG Loop



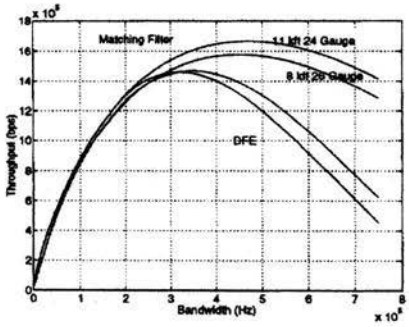
These estimations are very close to other study and measurement results (from referenced T1E1 contributions), as shown in Figure 14.5.

Figure 14.5 Receiver Performances Compared to the Channel Capacity



On the other hand, only on shorter than CSA loops—8 kft, 26-AWG and 11 kft, 24-AWG loops—a coding gain of a little higher 6 dB can be used in conjunction with a DFE-based transceiver to achieve a transmission throughput of 1.544 Mbps, as shown in Figure 14.6.

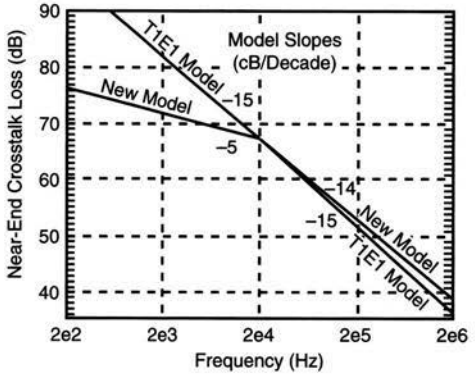
Figure 14.6 Receiver Performances on Shorter Loops



14.1.5 *Modifying the NEXT Model to Improve Performance*

The T1E1 contribution from ADTRAN[9] discusses the possibility of modifying the simplified crosstalk model and including the coupling transformer in the crosstalk model to obtain a simulated performance improvement of 0.8 dB. Specifically, the Unger (New) 49-disturber NEXT model should have a slope of –14 dB/decade instead of –15 dB/decade at frequencies above 20 kHz, as shown in Figure 14.7.

Figure 14.7 Unger and the Simplified NEXT Models



PairGain's T1E1 contribution[10] summarizes that the performance margins are about –2 dB with the simplified NEXT model and about –1 dB with the Unger NEXT model for uncoded systems. The Unger NEXT model has less crosstalk at lower frequencies, as shown in Figure 14.7. Table 14.10 summarizes SHDSL performance margins under different NEXT models.

Table 14.10 Performance Margin Summary on CSA Loop #6

Modulation	ADTRAN (Simplified/ Unger)	Level One (Unger)	AT&T (Simplified)	PairGain (Unger)
4-PAM	–2.6	–1.5	–2.1	–1.1
16-CAP	–3.1		–2.5	
32-CAP	–2.7		–2.1	–0.8 (5/2 PAM)
8-PAM	–2.6/–1.3	–1.7	–2.1	–1.3
64-CAP	–3.3		–2.6	

14.1.6 *Spectrum Compatibility*

A T1E1 contribution from AT&T studies the spectrum compatibility between SHDSL and ADSL.[11] CAP/QAM and PAM line codes are used for SHDSL. ADSL of 1.5 Mbps uses the 64-CAP line code and ADSL of 6 Mbps uses the DMT line code. Table 14.11 summarizes the results of SHDSL NEXT to the performance of a 1.5 Mbps ADSL. Table 14.12 summarizes the results of 1.5 Mbps CAP ADSL NEXT to the performance of an SHDSL. Table 14.13 summarizes the results of SHDSL NEXT to the performance of a 6 Mbps DMT ADSL. Results are expressed as ADSL or HDSL performance margins (above the SNR required for the 10^{-7} bit error rate).

Table 14.11 CAP ADSL Performance Margin Under SHDSL NEXT

	ADSL Margin (dB) (12 kft 24 AWG)	
HDSL Interferer	Ideal Spectrum	Practical Spectrum
64-CAP	8.9	6.9
128-CAP	18.0	14.8
8-PAM		-6.3
16-PAM	9.3	2.4

Table 14.12 SHDSL Performance Margin Under CAP ADSL NEXT

HDSL System	HDSL Margin (dB) for 9 kft, 26-AWG
64-CAP	15.1
128-CAP	18.6
8-PAM (no coding)	7.4
16-PAM	18.5
8-PAM	11.4

Table 14.13 DMT ADSL Performance Margin Under SHDSL NEXT

	ADSL Margin (dB) (12 kft 24 AWG)	
HDSL Interferer	Ideal Spectrum	Practical Spectrum
128-CAP	10.6	9.8
64-CAP	7.8	6.6
16-PAM	7.5	1.6

14.2 Searching for the Required Coding Gain

Many performance estimation studies for SHDSL have assumed a coding gain of about 4 dB. To meet the 6 dB performance margin requirement on CSA loops, a coding gain of about 6 dB is required. The requirement of such a high coding gain is a challenge both for the selection of a proper code and its implementation complexity.

14.2.1 Trellis Codes

A simple eight-state Trellis code can provide a coding gain of about 4 dB in conjunction with either 128-CAP or 16-PAM line codes. Trellis codes generally require a Tomlinson precoder and an adaptive transmitter instead of a Decision Feedback Equalizer (DFE).

Note
A Reed-Solomon code would also yield a 4 dB coding gain, but the resulting delay is too high.[12]